

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA INFORMÁTICA

Desarrollo de un sistema para el seguimiento de incendios forestales

Desarrollo de una herramienta de almacenamiento de información de
incendios forestales

Alumno: Javier Prieto Canela
Tutor: David Díez Cebollero
Versión 1.7 del documento
2 de Mayo de 2011

Resumen

Este documento contiene el Proyecto Fin de Carrera que tiene por objetivo la creación de una aplicación web que cumpla las siguientes características:

- Dotar de información espacio temporal a un conjunto de incidencias.
- Diferenciar y separar la información asociada a cada incidencia, dependiendo de su nivel de objetividad.
- Posibilidad de comparar de forma visual dos conjuntos de datos utilizando su información espacio/temporal, pudiendo ocultar o mostrar distintos tipos de incidencias.

Como objetivos secundarios se establecen los siguientes:

- Desarrollar una aplicación web robusta y adaptable con el fin de que pueda ampliarse o modificarse de forma sencilla y rápida.
- Tratar el dominio de la información de forma que pueda ser sustituido por otro distinto.

Para lograr estos objetivos se ha realizado un estudio y revisión de las distintas tecnologías que existen actualmente (**Revisión de tecnologías**). Tras su análisis se ha desarrollado una gestión real del proyecto (**Gestión de proyecto software**), definiendo el plan de trabajo a seguir, la gestión de recursos y riesgos y el plan de pruebas a ejecutar para validar el sistema construido.

A continuación se describe la solución que se va a adoptar (**Solución**) definiendo el proceso de desarrollo y, por último, mostrando el producto desarrollado. Seguidamente se procede a la evaluación del sistema implementado (**Evaluación**) mediante una batería de casos de prueba, para después poder realizar un análisis que concluya en la aceptación del sistema.

Como anexos al documento se dispone del seguimiento del proyecto (**Anexo II. Seguimiento de proyecto fin de carrera**) real y estimado, del manual de usuario con todas las instrucciones necesarias (**Anexo III. Manual de usuario**).

Abstract

This document contains the thesis that aims to create a web application that meets the following characteristics:

- Provide temporary space information to a set of incidents.
- distinguish and separate the information associated with each occurrence, depending on their level of objectivity.
- Ability to visually compare two sets of data using information space / time, can hide or display different types of incidents.

The secondary objectives are established as follows:

- Develop a robust and scalable Web application so that it can be expanded or modified easily and quickly.
- Treat the domain of information so that it can be replaced by another one.

To achieve these goals has undertaken a study and review of the different technologies that currently exist (***Revisión de tecnologías***). After analysis we have developed a real project management (***Gestión de proyecto software***), defining the roadmap to follow, the management of resources and risks and plan to run tests to validate the system built.

The following describes the solution to be taken (***Solución***) to define the development process and, finally, showing the product developed. It then proceeds to evaluate the system implemented (***Evaluación***) by a battery of test cases to later analysis to be completed by the acceptance of the system.

As annexes to the document are available for monitoring the project (***Anexo II. Seguimiento de proyecto fin de carrera***) real and estimated, the user manual with all necessary instructions (***Anexo III. Manual de usuario***).

Índice de contenidos

Resumen.....	3
Abstract	3
Índice de contenidos	5
Índice de figuras	6
Índice de tablas	8
Glosario de términos	14
1 Introducción.....	16
1.1 Planteamiento del problema	16
1.2 Objetivos	19
1.3 Metodología	21
1.4 Estructura de la memoria.....	21
2 Revisión de tecnologías.....	23
3 Gestión de proyecto software	36
3.1 Alcance del proyecto	36
3.2 Plan de trabajo	39
3.3 Gestión de recursos.....	40
3.4 Gestión de riesgos	41
3.5 Plan de pruebas.....	52
4 Solución.....	56
4.1 Descripción de la solución.....	56
4.2 El proceso de desarrollo.....	58
4.3 El producto del desarrollo	124
5 Evaluación	126
5.1 Proceso de evaluación.....	126
5.2 Análisis de resultados.....	138
6 Conclusión.....	140
6.1 Aportaciones realizadas	140
6.2 Trabajos futuros	140
6.3 Problemas encontrados	141
6.4 Opiniones personales.....	142
7 Bibliografía	143
Anexo I. Control de versiones	145
Anexo II. Seguimiento de proyecto fin de carrera	148
Anexo III. Manual de usuario	150
Anexo IV. Prototipo	166

Índice de figuras

Figura 1. Gráfica espacio-temporal.....	17
Figura 2. Concentración de incendios en España.....	18
Figura 3. Google Maps.	23
Figura 4. Ejemplo de GPolyline.	24
Figura 5. Polígono inicial para crear una figura geométrica.	26
Figura 6. Polígono modificado para crear una figura geométrica.....	26
Figura 7. Polígono final al modificar uno inicial.	27
Figura 8. Ejemplo Simile Timeline.	28
Figura 9. Línea temporal con Simile Timeline.	30
Figura 10. Ejemplo TimeMap - Ushahidi.	34
Figura 11. Diagrama de Gantt.	40
Figura 12. Organigrama.....	41
Figura 13. MVC + I.	80
Figura 14. Diagrama de subsistemas.....	81
Figura 15. Diagrama de clases general.....	82
Figura 16. Comunicación Vista-Controlador-Modelo.	83
Figura 17. Modelo de clases de la gestión de incidencias.....	84
Figura 18. Modelo de clases relacionado con la gestión de usuarios.	85
Figura 19. Relación capa Controlador con capa Modelo.	86
Figura 20. Modelo de clases de la gestión KML con la capa Infraestructura.	87
Figura 21. Modelo de clases de la capa Infraestructura.	96
Figura 22. Modelo de clases de la gestión de incidencias con la capa Infraestructura.	97
Figura 23. Modelo de clases de la gestión de usuarios con la capa Infraestructura.	97
Figura 24. Modelo de clases de la gestión de documentos KML con la capa Infraestructura.	98
Figura 25. Modelo de datos.	102
Figura 26. Organización del proyecto.	103
Figura 27. Lista de tareas del desarrollo.	104
Figura 28. Pantalla de inicio de la aplicación.	124
Figura 29. Pantalla General de la aplicación.	125
Figura 30. Pantalla de acceso al sistema.....	150
Figura 31. Pantalla general del sistema.	150
Figura 32. Menú Vistas.....	151
Figura 33. Menú Añadir.....	151
Figura 34. Menú Administración de Elementos.....	151
Figura 35. Menú Administración de Usuarios.....	151
Figura 36. Menú Ayuda.	151
Figura 37. Menú Salir del Sistema.	151
Figura 38. Menú Seleccionado.	151
Figura 39. Menú Deshabilitado.....	151
Figura 40. Mapa Vista General.....	152
Figura 41. Lista Vista General.....	153
Figura 42. Globo Mapa de la Vista General.....	154

Figura 43. Columna de Acciones.	154
Figura 44. Vista en Detalle - Acciones.	155
Figura 45. Dossier Vista de Detalle.....	156
Figura 46. Vista de Comparación.	157
Figura 47. Pantalla.....	158
Figura 48. Añadir Acción.	159
Figura 49. Añadir Reporte.	160
Figura 50. Controles Ubicación de Incidencia o Acción.	161
Figura 51. Polígono sin cerrar ni confirmar.....	162
Figura 52. Polígono cerrado y confirmado.....	162
Figura 53. Polígono relleno.	162
Figura 54. Punto sin confirmar.....	163
Figura 55. Administración de Incidencias, Acciones y Reportes.....	163
Figura 56. Administración de Usuarios.	164
Figura 57. Añadir Usuario.....	164
Figura 58. Pantalla de Login.	166
Figura 59. Pantalla General.	167
Figura 60. Pantalla Comparación.	168
Figura 61. Pantalla Guardar Incidencia.	169
Figura 62. Pantalla Guardar Acción.....	170
Figura 63. Pantalla Guardar Reporte.....	171
Figura 64. Pantalla Detalle de Acciones.	172
Figura 65. Pantalla Detalle de Reportes.....	173
Figura 66. Pantalla Administrar Incidencias, acciones y reportes.....	174
Figura 67. Pantalla Administrar Usuarios.....	175

Índice de tablas

Tabla 1. Glosario de términos.	15
Tabla 2. Número de incendios forestales entre 2004 y 2008 en España.	17
Tabla 3. Código de un evento con duración.	30
Tabla 4. Código de un evento sin duración.	31
Tabla 5. Código de un evento puntual.	31
Tabla 6. Código de carga de datos mediante JSON.	32
Tabla 7. Código para localizar dirección de descarga mediante JSON.	32
Tabla 8. Código para descargar datos mediante JSON.	32
Tabla 9. Código de carga de datos de un TimeMap.	34
Tabla 10. Salario bruto mensual.	37
Tabla 11. Coste de equipos y dispositivos.	37
Tabla 12. Coste de software.	38
Tabla 13. Costes de hosting y conectividad.	38
Tabla 14. Costes de material de oficina.	38
Tabla 15. Costes de implementación e instalación del producto.	38
Tabla 16. Costes totales.	38
Tabla 17. Cálculo total.	39
Tabla 18. Estimación de tareas.	40
Tabla 19. Fuentes de riesgos.	42
Tabla 20. Riesgo Terremoto.	42
Tabla 21. Riesgo Eléctrico.	43
Tabla 22. Riesgo Incendio.	43
Tabla 23. Riesgo Inundación.	43
Tabla 24. Riesgo Comunicación.	44
Tabla 25. Riesgo Expectativas.	44
Tabla 26. Riesgo Baja Temporal.	44
Tabla 27. Riesgo Sabotaje Interno.	44
Tabla 28. Riesgo Baja Permanente.	45
Tabla 29. Riesgo Baja del Jefe de Proyecto.	45
Tabla 30. Riesgo Baja de algún Responsable de Comité.	45
Tabla 31. Riesgo Modificación de Requisitos.	46
Tabla 32. Riesgo Sabotaje Externo.	46
Tabla 33. Riesgo Cancelación del Proyecto.	46
Tabla 34. Riesgo Espionaje.	46
Tabla 35. Riesgo Error en Requisito.	47
Tabla 36. Riesgo Saturación de Recursos.	47
Tabla 37. Riesgo Desviación/Error en la planificación.	47
Tabla 38. Riesgo Incompatibilidad de Versiones.	47
Tabla 39. Riesgo Fallos de Documentación.	48
Tabla 40. Riesgo Malestar en el Seno del Equipo.	48
Tabla 41. Riesgo Destrucción de las Infraestructuras de Trabajo.	48
Tabla 42. Riesgo Retrasos en la Entrega.	48

Tabla 43. Impacto de in riesgo en el desarrollo.	49
Tabla 44. Análisis cualitativo de los riesgos.	50
Tabla 45. Relación numérica del impacto.	50
Tabla 46. Factores de Riesgo individuales.....	51
Tabla 47. Cálculo del factor de riesgo medio.	51
Tabla 48. Pruebas unitarias.	53
Tabla 49. Pruebas de integración.	53
Tabla 50. Pruebas de implantación.	54
Tabla 51. Pruebas de aceptación.	54
Tabla 52. RE-CA-01 - Elementos del sistema.....	58
Tabla 53. RE-CA-02 – Consulta de incidencias.	58
Tabla 54. RE-CA-03 – Filtros en la consulta de incidencias.	59
Tabla 55. RE-CA-04 – Consulta de acciones.	59
Tabla 56. RE-CA-05 – Filtros en la consulta de acciones.	59
Tabla 57. RE-CA-06 – Consulta de reportes.	59
Tabla 58. RE-CA-07 – Filtros en la consulta de reportes.	59
Tabla 59. RE-CA-08 – Comparación de colección de incidencias.	60
Tabla 60. RE-CA-09 – Filtros en la comparación.....	60
Tabla 61. RE-CA-10 – Perfiles de usuario.	60
Tabla 62. RE-CA-11 – Perfil administrador.	60
Tabla 63. RE-CA-12 – Perfil validador.....	60
Tabla 64. RE-CA-13 – Perfil de usuario básico.....	61
Tabla 65. RE-CA-14 – Administración de incidencias.	61
Tabla 66. RE-CA-15 – Administración de usuarios.	61
Tabla 67. RE-CA-16 – Ayuda local.....	61
Tabla 68. RE-RE-01 – Idioma de la interfaz.	61
Tabla 69. RE-RE-02 – Utilización de Google Maps.	62
Tabla 70. RE-RE-03 – Utilización de Timemap.....	62
Tabla 71. RE-RE-04 – Lenguaje de programación Java.....	62
Tabla 72. RE-RE-05 – Servidor Apache Tomcat.	62
Tabla 73. RE-RE-06 – Base de datos MySQL.....	62
Tabla 74. RE-RE-07 – Navegadores válidos.	63
Tabla 75. RE-RE-08 – Log de estado del sistema.....	63
Tabla 76. RF-01 – Información asociada a una incidencia.	63
Tabla 77. RF-02 – Información asociada a una acción.	64
Tabla 78. RF-03 – Información asociada a un reporte.	64
Tabla 79. RF-04 – Información asociada a un usuario.	65
Tabla 80. RF-05 – Consulta de incidencias mediante un mapa.....	65
Tabla 81. RF-06 – Consulta de incidencias mediante una lista.	65
Tabla 82. RF-07 – Filtro de texto exclusivo en la consulta de incidencias.	65
Tabla 83. RF-08 – Filtro de texto inclusivo en la consulta de incidencias.	65
Tabla 84. RF-09 – Filtro de tipos en la consulta de incidencias.....	66
Tabla 85. RF-10 – Filtro por fecha en la consulta de incidencias.	66
Tabla 86. RF-11 – Consulta de acciones.	66
Tabla 87. RF-12 – Filtro de texto exclusivo en la consulta de acciones.	66
Tabla 88. RF-13 – Filtro de texto inclusivo en la consulta de acciones.	67
Tabla 89. RF-14 – Filtro por fecha en la consulta de acciones.	67

Tabla 90. RF-15 – Consulta de reportes.	67
Tabla 91. RF-16 – Filtro de texto exclusivo en la consulta de reportes.....	67
Tabla 92. RF-17 – Filtro de texto inclusivo en la consulta de acciones.	68
Tabla 93. RF-18 – Filtro por fecha en la consulta de acciones.	68
Tabla 94. RF-19 – Comparación de colección de incidencias.....	68
Tabla 95. RF-20 – Filtro de texto exclusivo en la comparación.....	68
Tabla 96. RF-21 – Filtro de texto inclusivo en la comparación.....	68
Tabla 97. RF-22 – Filtro de tipos en la comparación.....	69
Tabla 98. RF-23 – Filtro por fecha en la comparación.....	69
Tabla 99. RF-24 – Perfiles de usuario.	69
Tabla 100. RF-25 – Perfil administrador.....	69
Tabla 101. RF-26 – Perfil validador.	70
Tabla 102. RF-27 – Perfil de usuario básico.	70
Tabla 103. RF-28 – Guardar incidencia.	70
Tabla 104. RF-29 – Guardar acción.	70
Tabla 105. RF-30 – Guardar reporte.....	70
Tabla 106. RF-31 – Guardar ubicación.	70
Tabla 107. RF-32 – Administración de incidencias.....	71
Tabla 108. RF-33 – Administración de usuarios.	71
Tabla 109. RF-34 – Ayuda local.	71
Tabla 110. RD-01 – Datos de una incidencia.....	72
Tabla 111. RD-02 – Datos de una acción.....	72
Tabla 112. RD-03 – Datos de un reporte.....	72
Tabla 113. RD-04 – Datos de un usuario.....	73
Tabla 114. RI-01 – Comunicación de la aplicación con la base de datos.	74
Tabla 115. RO-01 – Comunicación del usuario con el sistema.....	74
Tabla 116. RO-02 – Ejecución del sistema.....	74
Tabla 117. RV-01 – Simulación de la aplicación.	75
Tabla 118. RPA-01 – Funcionamiento completo.....	75
Tabla 119. RD-01 – Manual de usuario.	75
Tabla 120. RSG-01 – Validación de campos.	75
Tabla 121. RSG-02 – Autenticación de usuarios.....	76
Tabla 122. RP-01 – Lenguaje de programación.....	76
Tabla 123. RP-02 – Base de datos.	76
Tabla 124. RP-03 – MySQL.	76
Tabla 125. RP-04 – Máquina virtual Java.	77
Tabla 126. RP-05 – Datos TimeMaps.....	77
Tabla 127. RP-06 – Codificación UTF-8.....	77
Tabla 128. RCAL-01 – Concurrencia.	77
Tabla 129. RM-01 – Log de estado del sistema.....	78
Tabla 130. RM-02 – Log errores internos.....	78
Tabla 131. RM-03 – Utilización de patrones en la implementación.	78
Tabla 132. Subsistema Vista.....	80
Tabla 133. Subsistema Controlador.	80
Tabla 134. Subsistema Modelo.	81
Tabla 135. Subsistema Infraestructura.	81
Tabla 136. Clase GestionIncidencia.....	88

Tabla 137. Clase GestionUsuarios.	89
Tabla 138. Clase GeneralKML.	89
Tabla 139. Clase DetailKML.	89
Tabla 140. Clase ComparisonKML.	90
Tabla 141. Clase AdminEventsKML.	90
Tabla 142. Clase ElementoNombreKML.	90
Tabla 143. Clase Incidencia.	90
Tabla 144. Clase SubIncidencia.	91
Tabla 145. Clase Recurso.	91
Tabla 146. Clase TipoRecurso.	91
Tabla 147. Clase Tipo.	91
Tabla 148. Clase SubTipo.	92
Tabla 149. Clase PuntosIncidencia.	92
Tabla 150. Clase PuntosSubincidencia.	92
Tabla 151. Clase Usuarios.	92
Tabla 152. Clase Perfil.	93
Tabla 153. Clase UserBean.	93
Tabla 154. Clase SubmitEventBean.	93
Tabla 155. Clase EditEventBean.	94
Tabla 156. Clase SubmitSubEventBean.	94
Tabla 157. Clase EditSubEventBean.	95
Tabla 158. Clase SubmitResourceBean.	95
Tabla 159. Clase EditResourceBean.	95
Tabla 160. Clase SubmitUserBean.	96
Tabla 161. Clase HibernateUtil.	99
Tabla 162. Clase HibernateUsuarios.	99
Tabla 163. Clase HibernatePerfil.	99
Tabla 164. Clase HibernateIncidencia.	99
Tabla 165. Clase HibernateSubIncidencia.	100
Tabla 166. Clase HibernateRecurso.	100
Tabla 167. Clase HibernateTipoSubTipo.	100
Tabla 168. Clase HibernateTipoRecurso.	101
Tabla 169. Clase HibernatePuntosIncidencia.	101
Tabla 170. Clase HibernatePuntosSubIncidencia.	101
Tabla 171. Herramientas utilizadas durante el desarrollo.	103
Tabla 172. PB-01-DAO.	105
Tabla 173. PB-02-DAO.	105
Tabla 174. PB-03-DAO.	106
Tabla 175. PB-04-DAO.	106
Tabla 176. PB-05-DAO.	107
Tabla 177. PB-06-DAO.	107
Tabla 178. PB-07-DAO.	108
Tabla 179. PB-08-DAO.	108
Tabla 180. PB-09-DAO.	108
Tabla 181. PN-01-DAO.	109
Tabla 182. PN-02-DAO.	109
Tabla 183. PB-03-DAO.	110

Tabla 184. PN-04-DAO.....	110
Tabla 185. PN-01-GESTION.	110
Tabla 186. PN-02-GESTION.	111
Tabla 187. PN-03-GESTION.	111
Tabla 188. PN-04-GESTION.	112
Tabla 189. PN-05-GESTION.	112
Tabla 190. PN-06-GESTION.	112
Tabla 191. PN-07-GESTION.	113
Tabla 192. PN-08-GESTION.	113
Tabla 193. PN-09-GESTION.	114
Tabla 194. PN-10-GESTION.	114
Tabla 195. PN-11-GESTION.	115
Tabla 196. PN-12-GESTION.	115
Tabla 197. PN-01-KML.....	116
Tabla 198. PN-02-KML.....	116
Tabla 199. PN-03-KML.....	116
Tabla 200. PN-04-KML.....	117
Tabla 201. PI-01.....	117
Tabla 202. PI-02.....	118
Tabla 203. PI-03.....	118
Tabla 204. PI-04.....	118
Tabla 205. PI-05.....	118
Tabla 206. PI-06.....	119
Tabla 207. PI-07.....	119
Tabla 208. PI-08.....	119
Tabla 209. PI-09.....	120
Tabla 210. PI-10.....	120
Tabla 211. PI-11.....	120
Tabla 212. PI-12.....	120
Tabla 213. PI-13.....	121
Tabla 214. PI-14.....	121
Tabla 215. PI-15.....	121
Tabla 216. PI-16.....	122
Tabla 217. PI-17.....	122
Tabla 218. PI-18.....	122
Tabla 219. PI-19.....	122
Tabla 220. PI-20.....	123
Tabla 221. Prueba P-01.	127
Tabla 222. Prueba P-02.	127
Tabla 223. Prueba P-03.	128
Tabla 224. Prueba P-04.	128
Tabla 225. Prueba P-05.	129
Tabla 226. Prueba P-06.	129
Tabla 227. Prueba P-07.	130
Tabla 228. Prueba P-08.	130
Tabla 229. Prueba P-09.	131
Tabla 230. Prueba P-10.	131

Tabla 231. Prueba P-11.	132
Tabla 232. Prueba P-12.	132
Tabla 233. Prueba P-13.	133
Tabla 234. Prueba P-14.	133
Tabla 235. Prueba P15.	133
Tabla 236. Prueba P-16.	134
Tabla 237. Prueba P-17.	134
Tabla 238. Prueba P-18.	135
Tabla 239. Prueba P-19.	135
Tabla 240. Prueba P-20.	135
Tabla 241. Prueba P-21.	135
Tabla 242. Prueba P-22.	136
Tabla 243. Prueba P-23.	136
Tabla 244. Prueba P-24.	136
Tabla 245. Prueba P-25.	136
Tabla 246. Prueba P-26.	137
Tabla 247. Prueba P-27.	137
Tabla 248. Prueba P-28.	137
Tabla 249. Análisis de los objetivos.	138
Tabla 250. Matriz de trazabilidad Pruebas-Requisitos Funcionales.	138
Tabla 251. Matriz trazabilidad Requisitos Funcionales-Pruebas de Evaluación.	139
Tabla 252. Versión 1.0 Documento.	145
Tabla 253. Versión 1.1 Documento.	145
Tabla 254. Versión 1.2 Documento.	145
Tabla 255. Versión 1.3 Documento.	145
Tabla 256. Versión 1.4 Documento.	146
Tabla 257. Versión 1.5 Documento.	146
Tabla 258. Versión 1.6 Documento.	147
Tabla 259. Versión 1.7 Documento.	147
Tabla 260. Planificación inicial del PFC.	148
Tabla 261. Planificación real del PFC.	149

Glosario de términos

Término	Descripción
AJAX	Asynchronous JavaScript And XML. Técnica de desarrollo web para crear aplicaciones interactivas. Mantiene comunicaciones asíncronas con el servidor en segundo plano.
Altova UModel	Herramienta de modelado de diagramas.
ASI	Análisis del sistema de información. Proceso de métrica V3.
Bean	Modelo de componentes para la construcción de aplicaciones en Java.
C3PO	Gestor de conexiones.
Case Studio	Herramienta de modelado de bases de datos.
Chrome	Navegador web desarrollado por Google.
CSI	Construcción del sistema de información. Proceso de métrica V3.
DAO	Data Access Object. Componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos.
Diagrama de Gantt	Herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas a lo largo de un tiempo determinado.
DSI	Diseño del sistema de información. Proceso de métrica V3.
Eclipse Helios	Entorno de desarrollo (IDE) de código abierto multiplataforma para desarrollar proyectos.
EVS	Estudio de viabilidad del sistema. Proceso de métrica V3.
Firefox	Navegador web desarrollado por Mozilla.
GeoRSS	Conjunto de estándares para representar información geográfica mediante el uso de capas. Construido dentro de la familia de estándares RSS.
Google	Motor de búsqueda propietario de Google Inc.
Google Maps	Servicio gratuito de Google Inc. Servidor de aplicaciones de mapas en la web.
Hibernate	Herramienta de mapeo objeto-relacional para la plataforma Java.
Hosting	Servicio que provee a los usuarios de internet un sistema para poder almacenar información accesible vía web.
IAS	Implantación y aceptación del sistema. Proceso de métrica V3.
IDE	Integrated Development Environment. Programa informático compuesto por un conjunto de herramientas de programación.
Internet Explorer	Navegador web desarrollado por Microsoft.
JAR	Tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java.
Java	Lenguaje de programación orientado a objetos.
JavaScript	Lenguaje de programación interpretado.
JSF	Java Server Faces. Tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario.
JSON	JavaScript Object Notation. Formato ligero para el intercambio de datos.
JSP	Java Server Pages. Tecnología Java que permite generar contenido dinámico para web.
KML	Keyhole Markup Language. Lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones.
Log4j	Biblioteca desarrollada por Java que permite elegir la salida y el nivel de granularidad de los mensajes a tiempo de ejecución.
Mashup	Aplicación o sitio web que utiliza recursos de más de una fuente para crear un servicio completo.

Métrica V3	Metodología de planificación, desarrollo y mantenimiento de sistemas de información.
MVC	Modelo-Vista-Controlador. Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control.
RSS	Really Simple Syndication. Formato XML para syndicar o compartir contenido web.
Safari	Navegador web desarrollado por Apple.
Servlet	Objetos java que corren dentro del contexto de un contenedor de servlets y extienden su funcionalidad.
Simile Timeline	API de línea temporal.
Switch	Dispositivo digital de lógica de interconexión de redes de computadoras.
Timeline	Línea temporal en el que se ordenan elementos cronológicamente.
Timemap	Elemento formado por un mapa y una línea temporal.
WAR	Web Application Archive. Archivo JAR utilizado para distribuir una colección de elementos que constituyen una aplicación web.
XML	Extensible Markup Language. Metalenguaje extensible de etiquetas desarrollado por W3C.

Tabla 1. Glosario de términos.

1 Introducción

A día de hoy la importancia que tiene el 'definir' la información requerida para llevar a cabo una correcta visualización y gestión, determinar qué fuentes de información se tienen que utilizar, la localización y acceso de esas fuentes y el correcto uso de la información es importante para llevar a cabo un proceso de prevención y actuación frente a una situación de emergencia como puede ser un incendio.

Se sabe que a través de la vista el hombre capta en torno al 80% de la información que recibe, por lo que es importante hacer hincapié en este sentido para mejorar la visualización de la información. Por tanto, parece razonable analizar y prestar más atención a la forma en que la información se presenta a un usuario. Se sabe que la forma adecuada de enfrentar y gestionar las situaciones de emergencia es un reto para el que resulta imprescindible estar bien preparado. Solamente una optimizada visualización de la información y una excelente gestión de la emergencia permitirán lograr un resultado favorable.

Además, consiguiendo una buena representación y visualización de toda la información relativa a un conjunto de incidencias, se pueden observar pautas o concentraciones que ayuden a prevenir estas situaciones, es decir, si se sabe que en una determinada zona se producen un gran número de incendios sería sensato prevenir antes que tener que actuar. Cuando por cualquier otro motivo la prevención resulta imposible entra en juego una gestión basada en el control y asignación de recursos. Una vez que se tiene una excelente organización y visualización de la información, resulta más fácil lograr una mejor gestión en una situación de emergencia.

1.1 Planteamiento del problema

Existen numerosas herramientas que permiten gestionar una incidencia y, en cierta medida, presentar la información de forma adecuada para que el usuario la pueda localizar de forma rápida. Si se entra en el campo de la prevención de incidentes, estas herramientas pueden no resultar adecuadas si no se tiene una buena visualización de toda la información, ya sea de una incidencia o de un número elevado de éstas.

Para plantear los problemas que motivan este proyecto hay que contestar las siguientes cuestiones:

- Necesidad de diferenciar información objetiva y subjetiva.
- Cantidad de información a la que hay que enfrentarse.
- Representación óptima de información espacio temporal.

La información objetiva es aquella que no tiene manipulación y transmite exactamente lo que dice. Sin embargo, la información subjetiva puede ser manipulada transformándola en tendenciosa según el punto de vista del emisor. Es importante separar estos canales de comunicación para mantener la integridad de la información, y de esta manera poder tomar decisiones correctas a la hora de realizar una gestión o prevención en el dominio de este problema, los incendios forestales. La información que transmite la fecha o el lugar es objetiva y la información que transmite una fotografía puede influir en la opinión de la gente dependiendo del título, la perspectiva, etc., es decir, puede ser subjetiva.

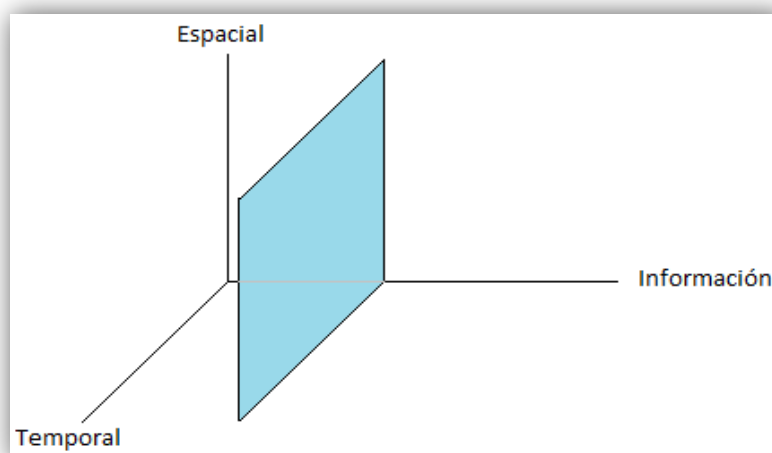


Figura 1. Gráfica espacio-temporal.

Observando la *Figura 1. Gráfica espacio-temporal*, se pueden ver las dimensiones *Espacial*, *Temporal* e *Información*. Las dimensiones *Espacial* y *Temporal* representan el tiempo y el espacio de cada elemento de información, entendiendo por información datos objetivos o subjetivos relativos a una incidencia. El eje *Información* se refiere a la objetividad o subjetividad de la información. Una información será objetiva si el valor de la dimensión *Información* es cero. En cambio, será subjetiva cuanto mayor sea el valor del eje *Información*.

Un elemento asociado a una incidencia tendrá un valor para el eje *Espacial* que corresponderá con el lugar dónde fue tomado. El valor de la coordenada *Temporal* será el instante de tiempo en el que fue tomado. Y el valor de la dimensión *Información* corresponderá con lo objetivo o subjetivo que es la información.

Por lo tanto, es importante distinguir que canales de información son formales u objetivos y que canales de información son informales o subjetivos.

La cantidad de información a la que se puede enfrentar un usuario a la hora de tratar con incendios forestales puede ser grande, y más aún si se tienen en cuenta varios años atrás. Es importante intentar que el número de incidencias no influya a la hora de localizar alguna de ellas.

La cantidad de datos a la que este proyecto se puede enfrentar se documenta con la siguiente Tabla 2, en la que se indica el número de incendios producidos en España en los últimos años [1]:

	Año 2004	Año 2005	Año 2006	Año 2007	Año 2008
Nº Incendios	21.396	25.492	16.334	10.932	11.612

Tabla 2. Número de incendios forestales entre 2004 y 2008 en España.

El problema de la representación de los datos espacio-temporales está relacionado con la información que indica la fecha y el lugar en el que se ha producido una incidencia. Tener que leer la información espacio-temporal de cada incidencia es muy laborioso. Además, la forma de representar el lugar en el que se ha producido la incidencia se puede volver muy dura si el área está formada por un conjunto de áreas o si abarca a uno o más municipios. El problema de manejar el aspecto temporal viene cuando se dispone de un gran número de incidencias.

Averiguar que sucesos del dominio del sistema han ocurrido en un lugar y tiempo determinado o el área que han abarcado, etc. traería mucho gasto de tiempo si se tiene que comprobar informe a informe, y más si dentro de cada informe hay comprobar si la o las direcciones del evento corresponden con los parámetros de búsqueda.

Este problema se justifica si se tiene en cuenta el artículo *“Spatiotemporal Mashups: A Survey of Current Tools to Inform Next Generation Crisis Support”* que Sophia Liu y Leysia Palen han publicado [2]. En este artículo se indica que la información espacio-temporal es un aspecto importante de la información en una situación de emergencia, por lo que se puede deducir que una buena representación de la información espacio-temporal es prioritaria para conseguir una buena gestión.

En la siguiente frase *“mostrar ciertos tipos de datos temporales a través de un mapa puede proporcionar visualizaciones ricas”* [3], se puede apreciar que es un inconveniente quedarse en la presentación textual y no llevar la información a otro tipo o nivel de representación. Esta afirmación queda respaldada con la Figura 2 [4]:

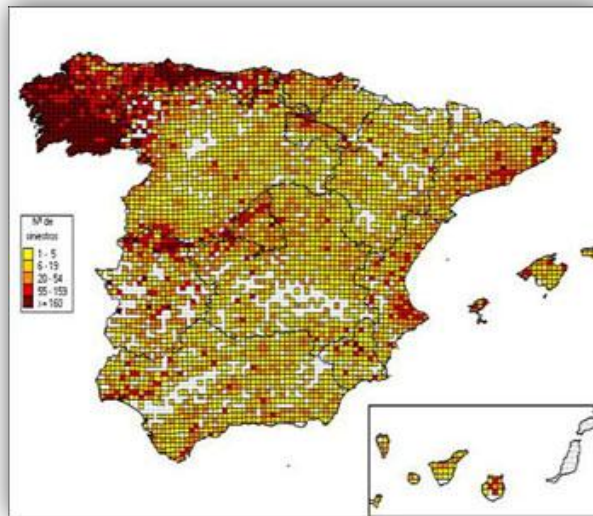


Figura 2. Concentración de incendios en España.

En el mismo artículo se muestra la importancia de tener la información ordenada cronológicamente para que el usuario pueda avanzar o retroceder en el tiempo y se pueda ver el orden en el que se van produciendo los distintos incidentes. También hay que destacar el problema que Sophia Liu y Leysia Palen mencionan: *“si demasiados informes se refieren al mismo lugar y/o al mismo instante de tiempo, podría hacer el mashup inútil debido a la confusión causada por la acumulación caótica de la información en un punto.”* [2] Esta cuestión es muy importante porque se pueden acumular una gran cantidad de datos en torno a un lugar y/o mismo instante, perjudicando la manera de visualización de la información.

La aplicación SIU6 [5] se encarga de almacenar y gestionar datos e informes sobre distintos tipos de incidencias. Las incidencias que gestiona esta herramienta son, entre otras, incendio de vegetación, incendio urbano o industrial y NRBQ.

Por cada tipo de incidencia se almacenan datos como los servicios que intervinieron, el número y tipo de víctimas que hubo, los recursos que se utilizaron, etc. Las incidencias del tipo incendio urbano o industrial tienen una característica en común. La característica que comparten es la localización, en el caso de que se den, ocurrirán en una dirección determinada. Como mucho podría abarcar varias direcciones.

Sin embargo, las incidencias del tipo incendio de vegetación y NRBQ pueden darse en un área determinada, y es difícil especificar dicha área utilizando direcciones convencionales. Un ejemplo muy concreto es una incidencia del tipo incendio de vegetación. Si el incendio se da en una superficie que comprende a varios municipios, sería más fácil asignar un área dónde se haya producido el incendio en vez de una dirección. Actualmente, en el sistema SIU6, sólo se introduce una dirección si se produce una incidencia de este tipo. En el caso de que se quiera indicar un área mayor se puede introducir la provincia. Pero si el incendio afecta a varias provincias el sistema no tienen una solución bien definida.

El sistema SIU6 no permite ver dónde se han producido los incendios, sólo aporta una dirección que puede resultar poco precisa o requerir del uso de un mapa para poder situarlo. Tampoco hace filtros en el que se puedan escoger los incendios por fechas y así poder ver dónde se concentran o si siguen alguna tendencia. En cuanto a tendencia se quiere decir si, a medida que pasa el tiempo, los incendios se van desplazando en alguna dirección. Este dato sería muy importante para poder destinar unidades y cubrir zonas que tuvieran una alta probabilidad de presentar este tipo de incidencias.

Si se quisiera averiguar si los incendios que se producen siguen alguna tendencia habría que estudiar cada informe y realizar un estudio que podría llevar mucho tiempo de trabajo. Del mismo modo, tampoco se puede observar si la frecuencia de incendios en una determinada zona está aumentando o decreciendo.

Esta herramienta es útil para comprender la información que se puede necesitar, anticiparse a los problemas de representación de la información que pueden surgir, como por ejemplo la representación de un área, y la inclusión de ciertas funcionalidades que se creen necesarias.

Habiendo contestado a las cuestiones anteriores se indican los problemas que han motivado este proyecto:

- Representar de forma correcta información espacio temporal.
- Necesidad de separar la distinta información que llegan a través de distintos canales de comunicación.
- Dificultad a la hora de realizar consultas textuales sobre datos visuales, como puede ser la ubicación de un evento.

1.2 Objetivos

Existe un problema si se pretende representar de forma textual los datos espaciales, debido a que resulta difícil especificar el área exacta en el que se produce un incendio forestal. Tener que acceder a la información espacio-temporal de cada incidencia de forma textual es muy laborioso ya que puede ser poco específica o contener demasiada información. Por lo que el problema de manejar el aspecto temporal viene cuando se dispone de un gran número de incidencias.

La información espacio-temporal es un aspecto importante de la información en una situación de emergencia, por lo que se puede deducir que una buena representación de la información espacio-temporal es prioritaria para conseguir una buena gestión.

“Mostrar ciertos tipos de datos temporales a través de un mapa puede proporcionar visualizaciones ricas” [6], esto indica que es un inconveniente quedarse en la presentación textual y no llevar la información a otro tipo o nivel de representación.

Por lo tanto, el objetivo principal de este proyecto es desarrollar una aplicación web o mashup que permita visualizar cualquier número de incidencias de tipo incendio forestal junto con toda la información espacio temporal con la que esté relacionada mediante una visualización rica.

Se plantean los siguientes objetivos:

- Dotar de información espacio temporal a un conjunto de incidencias:
 - Aportar de forma visual la información espacial de todas las incidencias colectiva o individualmente.
 - Aportar de forma visual la información temporal de todas las incidencias colectiva o individualmente.
 - Facilitar la búsqueda y localización de incidencias mediante su información espacio temporal.
- Diferenciar y separar la información asociada a cada incidencia:
 - Identificar la información objetiva relacionada con cada incidencia incluida en las distintas acciones.
 - Identificar la información subjetiva relacionada con cada incidencia incluida en los distintos reportes.
 - Facilitar la búsqueda y localización de las acciones y reportes utilizando su información espacio temporal.
- Posibilidad de comparar de forma visual dos conjuntos de datos utilizando su información espacio/temporal, pudiendo ocultar o mostrar tipos de incidencias.

Como objetivos secundarios se establecen los siguientes:

- Desarrollar la aplicación web robusta y adaptable con el fin de que pueda ampliarse o modificarse de forma sencilla y rápida.
- Tratar el dominio de la información de forma que puedan ser sustituidos por otro dominio.

Realizando el diseño de una interfaz teniendo en cuenta que cierta información tiene que ser representada de forma visual es un acierto. Este aspecto influirá a la hora de realizar búsquedas.

Además, consiguiendo una buena representación y visualización de toda la información relativa a un conjunto de incidencias, se pueden observar pautas o concentraciones que ayuden a prevenir estas situaciones, es decir, si se sabe que en una determinada zona se producen un gran número de incendios sería sensato prevenir antes que tener que actuar. Cuando por cualquier otro motivo la prevención resulta imposible entra en juego una gestión basada en el control y asignación de recursos.

Una vez que se tiene una excelente organización y visualización de la información, resulta más fácil conseguir una mejor gestión o prevención de una situación de emergencia. Este proyecto se va a centrar en la prevención pudiendo observar patrones en el conjunto de incidencias. También se va a centrar en el tipo de información que se asocia a cada incidencia, y así poder clasificarla según su objetividad o formalidad.

Por tanto, la aportación fundamental de este proyecto será la implementación de una herramienta que permita visualizar la información espacio temporal de un conjunto de

incidencias, diferenciar la distinta información objetiva y subjetiva asociada a cada una y realizar comparaciones visuales de dos conjuntos de incidencias.

1.3 Metodología

En esta sección se describe brevemente la metodología empleada en la realización del proyecto, así como el método de evaluación utilizado para validar las pruebas realizadas.

Estudio y determinación del problema

- Descripción del contexto del problema: definición del contexto del problema y los motivos que dan origen al desarrollo del proyecto.
- Revisión de tecnologías: estudio de las distintas tecnologías existentes con el fin de poder representar información de incendios forestales.
- Definición del problema: se enumerarán todos los problemas encontrados, siendo la solución a estos, parte importante de los requisitos del sistema a desarrollar.

Construcción de la solución

- Descripción de la solución que se pretende desarrollar, indicando los requisitos principales que el sistema deberá implementar.
- Análisis de la solución: enumeración detallada de todos los requisitos necesarios para satisfacer los objetivos propuestos.
- Diseño de la solución: realización del diseño de la futura aplicación web teniendo en cuenta todos los requisitos extraídos en la fase de análisis.
- Implementación: realización del diseño construido.
- Pruebas: enumeración del conjunto de pruebas que verifican la inclusión de todos los requisitos definidos.

Evaluación

Para llevar a cabo la validación de la aplicación web se empleará un plan de pruebas con el fin de demostrar:

- Alcance de los objetivos planteados.
- Inclusión de todos los requisitos extraídos en el sistema construido.

Análisis de los resultados

En esta fase del proyecto se expondrán las conclusiones alcanzadas una vez realizo el plan de pruebas descrito.

Para obtener resultados tangibles se construirán una serie de matrices de trazabilidad para averiguar si todos los objetivos y requisitos descritos han sido tenidos en cuenta.

1.4 Estructura de la memoria

En esta sección se expone la estructura del documento construido junto con el objetivo de cada capítulo.

- Capítulo 1: Introducción al trabajo, resumen de los objetivos y descripción de la metodología de trabajo.
- Capítulo 2: Estudio de las diferentes tecnologías disponibles.

- Capítulo 3: Definición del alcance del proyecto, tareas, recursos y riesgos.
- Capítulo 4: Descripción de la solución, desarrollo y producto desarrollado.
- Capítulo 5: Evaluación de la solución propuesta.
- Capítulo 6: Descripción de las conclusiones obtenidas, aportaciones realizadas, líneas de trabajo futuras, problemas encontrados a lo largo del tiempo de desarrollo y opiniones personales del trabajo.
- Capítulo 7: Bibliografía utilizada para la investigación y desarrollo.
- Anexo I: Seguimiento de las distintas versiones de la memoria.
- Anexo II: Seguimiento de las distintas tareas que se han llevado a cabo.
- Anexo III: Elaboración del manual de usuario de la aplicación desarrollada.
- Anexo IV: Descripción de las principales pantallas de la aplicación web.

2 Revisión de tecnologías

En este apartado se van a exponer las diferentes tecnologías que se considerarán para ser utilizadas en el proyecto. Las principales tecnologías estudiadas son:

- Google Maps.
- Timeline.
- Timemap.
- Mashups.

Este análisis de herramientas se utilizará como forma de encaminar la solución.

Google Maps

Google Maps¹ [7] es un servicio gratuito de Google. Este servicio ofrece imágenes de mapas desplazables, fotos de satélite y rutas entre diferentes ubicaciones. Hay que señalar que Google Maps es un mashup de tipo consumidores, es decir, reúne una gran cantidad de datos para que el usuario los utilice a través de una interfaz simple. En la *Figura 3. Google Maps.* se muestra un ejemplo de este servicio.



Figura 3. Google Maps.

Google ofrece una API para utilizar el servicio Google Maps de forma gratuita pero sin realizar más de 500.000 peticiones diarias. Google Maps API² proporciona diversas utilidades para manipular Google Maps y añadir contenido adicional mediante diversos servicios, permitiendo

¹ Definición Google Maps: http://es.wikipedia.org/wiki/Google_Maps.

² Definición Google Maps API: <http://code.google.com/intl/es-ES/apis/maps/index.html>.

crear mashups totalmente personalizados. Para poder utilizar Google Maps se necesita pedir una Key o clave³ que se asociará al dominio donde se aloje la página web.

Es importante saber que Google Maps está muy relacionado con JavaScript, por lo que habrá que tener en cuenta:

- Cualquier usuario que necesite utilizar este servicio deberá permitir la ejecución de esta tecnología.
- El programador deberá tener en cuenta esta tecnología a la hora de diseñar el sistema.

Google Maps API permite realizar multitud de acciones sobre Google Maps. Una de las principales acciones que Google Maps permite es ubicar una serie de iconos o polígonos en el mapa. Cada punto es representado mediante la clase *GLatLng* de Google Maps API. La clase *GLatLng* recibe como parámetros la latitud y longitud del punto que se quiere representar, por lo que resultaría difícil introducir nuevos puntos sin conocer sus coordenadas. Para ello, Google Maps API permite determinar que coordenadas pertenecen a una ubicación utilizando las siguientes funciones de la clase *GEvent*:

- *mousemove*: devuelve las coordenadas del punto sobre el que esté el puntero del ratón.
- *mouseover*: devuelve las coordenadas del punto sobre el que se haga clic.

Utilizando las anteriores funciones mediante Java Script, ya que Google Maps trabaja con Java Script, resultaría muy sencillo situar nuevos puntos sobre Google Maps en tiempo de ejecución ya que se podrían obtener las coordenadas librando al usuario de dicha tarea.

Google Maps API, además de situar iconos en el mapa, también permite dibujar figuras geométricas mediante la clase *GPolyline*. Para representar cualquier figura sólo hay que crear una serie de puntos donde cada uno será un vértice de la figura. Una vez que se tiene la figura sólo hay que añadir la capa de la figura a Google Maps obteniendo un resultado como el que se muestra en la Figura 4. Ejemplo de *GPolyline*.



Figura 4. Ejemplo de GPolyline.

³ Formulario para pedir Key de Google Maps: <http://code.google.com/intl/es-ES/apis/maps/signup.html>

Para este ejemplo se han utilizado seis puntos aplicando la clase *GLatLng* de Google Maps API. Para cerrar la figura, el último punto debe ser igual que el primero. Los parámetros configurables de *GPolyline* son los siguientes:

- Color del segmento: determina el color de los segmentos del polígono. Utiliza colores en formato *RGB*⁴.
- Grosor del segmento: determina el grosor de los segmentos del polígono. A mayor valor mayor grosor. El valor de esta propiedad no depende del zoom de Google Maps, por lo que el grosor del segmento es constante a cualquier zoom aplicado.
- Opacidad del segmento: determina lo opaco que serán los segmentos del polígono. Es un valor que oscila entre cero, totalmente diáfano, a uno, totalmente opaco.

En el anterior ejemplo que corresponde con la *Figura 4. Ejemplo de GPolyline*, los parámetros utilizados fueron:

- Color del segmento: #0000dd
- Grosor del segmento: 6
- Opacidad del segmento: 0.7

En este ejemplo los datos se cargaron mediante código JavaScript. Para que el usuario pueda construir el polígono rápidamente y de una manera sencilla se puede utilizar la clase *GPolyline* y editar los puntos que forman el polígono en tiempo real utilizando también JavaScript. Existen dos caminos para realizar esta tarea:

- Editar y desplazar los puntos de una figura ya creada mediante el ratón.
- Construir el polígono punto por punto utilizando el puntero del ratón.

Cuando se editan los puntos de una figura ya creada puede ocurrir que existan una gran cantidad de puntos muy próximos y esto dificulte la creación del polígono. Aun así es una forma muy sencilla si se aplica un zoom adecuado en la zona de Google Maps dónde se quiera crear el polígono. El primer paso del proceso sería situar el polígono inicial en el mapa tal y como se muestra en la *Figura 5. Polígono inicial para crear una figura geométrica.*:

⁴ Definición del formato de colores RGB. <http://es.wikipedia.org/wiki/RGB>



Figura 5. Polígono inicial para crear una figura geométrica.

Seguidamente, se desplazará cada vértice o puntos de mitad de segmento a la posición deseada. Por cada segmento recto que se forme, se creará un punto en la mitad que se podrá utilizar para definir de una manera más precisa la figura final:



Figura 6. Polígono modificado para crear una figura geométrica.

Por último, si la figura que se ha creado es la que se está buscando, se generará el polígono definitivo capturando, mediante código JavaScript, todas las coordenadas de los vértices del polígono. El resultado final sería el siguiente:



Figura 7. Polígono final al modificar uno inicial.

La segunda manera de crear figuras es más sencilla y consiste en colocar con el puntero del ratón haciendo clic cada punto. El proceso finaliza cuando el usuario coloca el último punto sobre el primero, consiguiendo un polígono como el de la *Figura 7*.

Ambas formas producen el mismo resultado, pero el segundo puede tener un uso más sencillo e intuitivo para cualquier usuario.

Esta tecnología se utilizará para situar elementos la dimensión espacio.

Timeline

Timeline⁵ [8] es una tecnología que permite ordenar distintos elementos en una línea temporal, pudiendo avanzar o retroceder de distintas maneras. En la *Figura 8* se puede ver que la línea temporal está dividida horizontalmente en dos partes. En la parte superior se sitúan los eventos. La parte inferior se utiliza para avanzar o retroceder en el tiempo.

⁵ Página home Timeline: <http://www.simile-widgets.org/timeline/>

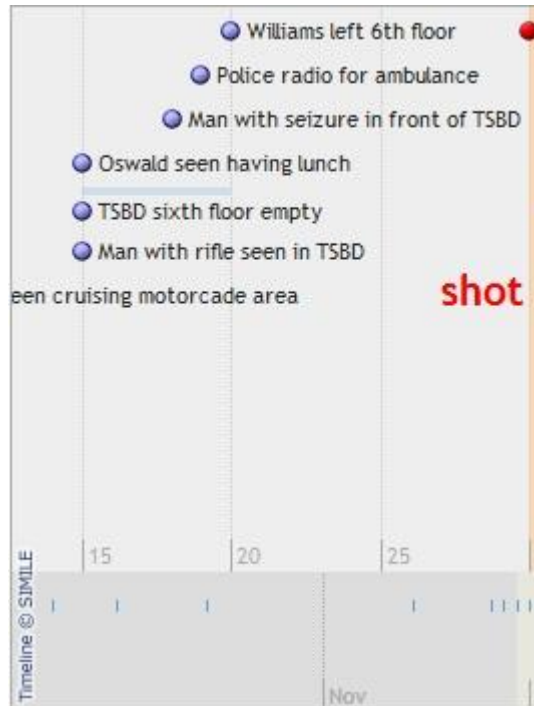


Figura 8. Ejemplo Simile Timeline.

Existen multitud de tipos de líneas temporales. Unas tienen en cuenta la hora exacta en la que se produjo un evento, otras tienen en cuenta el día y otras sólo consideran el año en el que ocurrió un acontecimiento. En el caso que se está estudiando se ha tenido en cuenta el día en el que se produjo suceso, por lo que se puede avanzar o retroceder en la línea temporal de día en día.

Para poder utilizar esta tecnología en una aplicación, será necesario importar las librerías⁶ que Simile Timeline proporciona.

Una vez conocido qué es y cómo se utiliza una línea temporal, se estudiará a fondo qué información es necesaria para crear elementos y añadirlos a la línea temporal. Para introducir elementos en una línea temporal, Simile Timeline puede utilizar los siguientes formatos:

- XML⁷: mediante un fichero XML.
- JSON⁸: alternativa al uso de XML en AJAX. Más ligero que XML pero menos seguro.
- SPARQL⁹: lenguaje de consulta para utilizar dentro de una página web semántica.

Carga de datos mediante XML

La utilización de XML es muy eficaz cuando se utiliza una línea temporal en local y no es necesario realizar ninguna conexión a alguna fuente externa.

Las etiquetas del documento XML para la carga de datos en una línea temporal Simile Timeline son:

⁶ Descargas de Simile Timeline: <http://code.google.com/p/simile-widgets/downloads/list>

⁷ XML. <http://es.wikipedia.org/wiki/XML>

⁸ JSON. <http://www.es.wikipedia.org/wiki/JSON>

⁹ SPARQL. <http://es.wikipedia.org/wiki/SPARQL>

- `<data></data>`: entre estas etiquetas se encuentran todos los eventos que se representarán en la línea temporal.
- `<event></event>`: entre estas etiquetas se encuentra la descripción de cada evento de la línea temporal. Se puede utilizar HTML para dar formato al texto.

Como se puede ver sólo existen dos tipos de etiquetas, pero la etiqueta `<event></event>` contiene una serie de parámetros para dotar de más información a cada evento. Los parámetros son:

- `start`: define la fecha en la que se inicia el evento.
- `end`: define la fecha en la que finaliza el evento. Tiene que tener el mismo formato que la fecha de inicio.
- `latestStart`: define una fecha de inicio aproximada. Tiene que tener el mismo formato que la fecha de inicio.
- `earliestEnd`: define una fecha de fin aproximada. Tiene que tener el mismo formato que la fecha de inicio.
- `durationEvent`: indica si el evento se prolonga en el tiempo. Los valores que acepta son `true` y `false`. En el caso que se utilice el valor `true`, entre las fechas de inicio y fin se dibujará un rectángulo indicando la duración del evento. En el caso de utilizar el valor `false` sólo se dibujará un punto en la línea temporal.
- `title`: define el título que aparecerá en la línea temporal.
- `icon`: imagen que aparecerá junto al título en la línea temporal.
- `image`: define el icono que aparecerá en la descripción del evento.
- `link`: define la URL a la que hará referencia el título del evento.
- `color`: color del texto del evento en la línea temporal.
- `textColor`: color del texto de la ventana de descripción del evento.

Los formatos de fechas admitidas son:

- Formato GMT: May 10 1961 00:00:00 GMT-0600
- Formato RFC 2822: Thu, 21 Dec 2000 16:01:07 +0200
- Objeto Date (con JSON): `new Date(Date.UTC(2008,0,17,20,00,00,0))`
- Formato ISO 8601: 1995-02-04 10:20:01Z

Existen unos formatos de fechas que no se recomiendan porque pueden resultar ambiguos. El problema es que se puede tener una configuración local de la máquina, y por tanto se puede interpretar de dos formas una misma fecha. Por ejemplo, la fecha "5/10/1961 00:00:00 GMT-0600" puede tener dos interpretaciones, día 10 de mayo o día 5 de octubre. Otro ejemplo a tener muy en cuenta es el siguiente: "5 10 1961 00:00:00 GMT-0600". En este caso los navegadores Firefox, Safari y Chrome interpretan la fecha correctamente, sin embargo el navegador Internet Explorer no lo hace. Es importante definir bien el formato de fecha que se va a utilizar para evitar problemas de compatibilidad y optimizar el rendimiento de la aplicación.

Un ejemplo de uso de Simile Timeline es el siguiente:

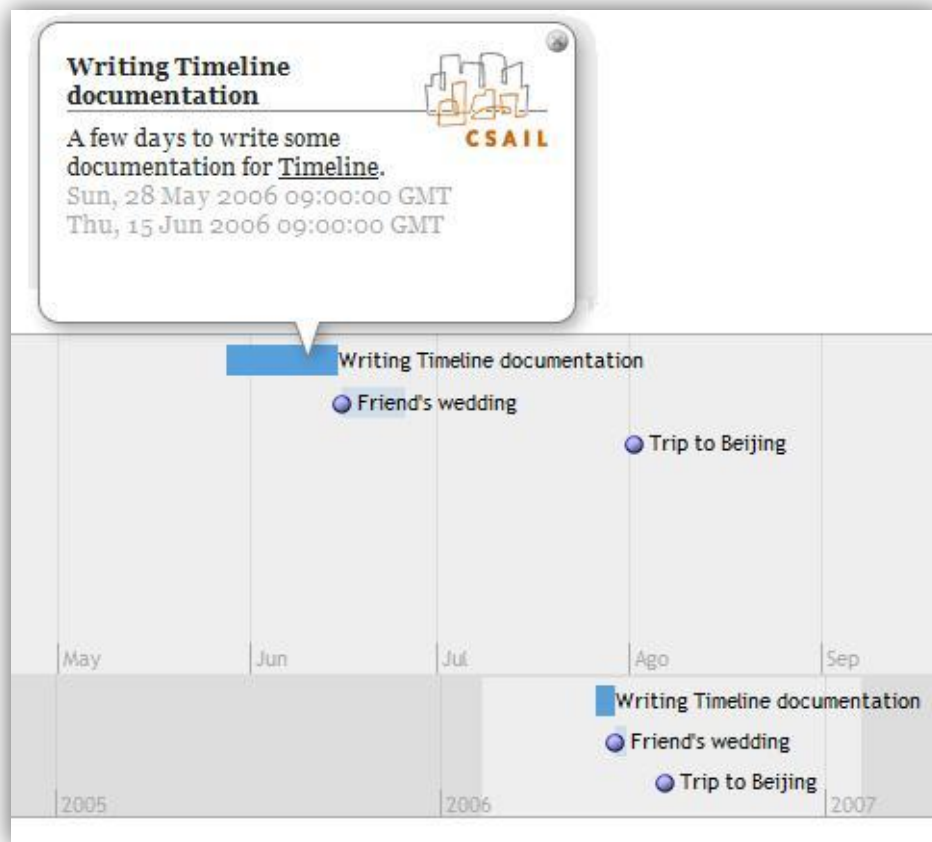


Figura 9. Línea temporal con Simile Timeline.

A continuación se explica que se ha utilizado para representar cada evento de la línea temporal:

- El formato para representar las fechas ha sido GMT.
- Los datos se han cargado de un documento XML.
- Evento “*Writing Timeline documentation*”: dado que su atributo *durationEvent* tiene el valor *true*, Simile Timeline extiende desde la fecha inicial a la fecha final un rectángulo azul, queriendo transmitir que el evento no es puntual y se prolonga en el tiempo. Al hacer clic en un evento de la línea temporal aparece una ventana que contiene una pequeña descripción, la fecha de inicio, la fecha de fin y una imagen que aparece en la esquina superior derecha. El código XML de este evento se representa en la *Tabla 3*:

```
<event start="May 28 2006 09:00:00 GMT" end="Jun 15 2006 09:00:00 GMT" durationEvent="true"
  title="Writing Timeline documentation" image="http://simile.mit.edu/images/csail-logo.gif">
```

A few days to write some documentation for Timeline.

```
</event>
```

Tabla 3. Código de un evento con duración.

- Evento “*Friend’s Wedding*”: en este caso, el atributo *durationEvent* tiene el valor *false*, por lo que sólo aparece un punto en la línea temporal. Aun así, como el evento tiene una fecha

inicial y una fecha final, Simile Timeline representa este intervalo de tiempo con un rectángulo azul pero muy diáfano. El código XML de este evento se representa en la *Tabla 4*:

```
<event start="Jun 16 2006 00:00:00 GMT" end="Jun 26 2006 00:00:00 GMT" title="Friend's wedding">  
  
    I'm not sure precisely when my friend's wedding is.  
  
</event>
```

Tabla 4. Código de un evento sin duración.

- Evento *"Trip to Beijing"*: este es el caso más sencillo de evento que Simile Timeline representa. Este evento sólo tiene una fecha inicial, por lo que será representado por un punto en la línea temporal. Además, al utilizar el atributo link, el título de la ventana de descripción será un enlace a la página web indicado en el XML que define al evento. El código XML de este evento se representa en la *Tabla 5*:

```
<event start="Aug 02 2006 00:00:00 GMT" title="Trip to Beijing" link="http://travel.yahoo.com/">  
  
    Woohoo!  
  
</event>
```

Tabla 5. Código de un evento puntual.

Carga de datos mediante JSON

JSON (JavaScript Object Notation), es un formato ligero para el intercambio de datos. JSON es un subconjunto de objetos de JavaScript que no requiere el uso de XML. Sirve como alternativa a XML en AJAX.

La utilización de JSON es muy eficaz cuando el volumen de los datos que la línea temporal va a consumir es grande y cuando puede haber un número de usuarios considerable utilizando el sistema. Primero hay que definir el formato que tendrán los datos y luego descargarlos en el cliente para su utilización.

Los parámetros de los datos con JSON son:

- start: define la fecha en la que se inicia el evento.
- end: define la fecha en la que finaliza el evento. Tiene que tener el mismo formato que la fecha de inicio.
- latestStart: define una fecha de inicio aproximada. Tiene que tener el mismo formato que la fecha de inicio.
- earliestEnd: define una fecha de fin aproximada. Tiene que tener el mismo formato que la fecha de inicio.
- durationEvent: indica si el evento se prolonga en el tiempo. Los valores que acepta son true y false. En el caso que se utilice el valor true, entre las fechas de inicio y fin se dibujará un rectángulo indicando la duración del evento. En el caso de utilizar el valor false sólo se dibujará un punto en la línea temporal.
- title: define el título que aparecerá en la línea temporal.
- icon: imagen que aparecerá junto al título en la línea temporal.
- image: define el icono que aparecerá en la descripción del evento.
- link: define la URL a la que hará referencia el título del evento.

- color: color del texto del evento en la línea temporal.
- textColor: color del texto de la ventana de descripción del evento.
- description: descripción del evento que se mostrará al hacer clic sobre él.

Un ejemplo de carga de datos con JSON se representa en la *Tabla 6*:

```
{
  'wiki-url':"http://simile.mit.edu/shelf/",
  'wiki-section':"Simile JFK Timeline",
  'dateTimeFormat': 'Gregorian',
  'events': [
    {
      'start':"Sat May 20 1961 00:00:00 GMT-0600",
      'title':"Bay of Pigs' Invasion",
      'durationEvent':false
    }, {
      'start':"Wed May 01 1963 00:00:00 GMT-0600" ,
      'end':"Sat Jun 01 1963 00:00:00 GMT-0600" ,
      'durationEvent':true,
      'title':"Oswald moves to New Orleans",
      'description':"Oswald moves to New Orleans, and finds employment at the
William B. Riley Coffee Company. <i>ref. Treachery in Dallas, p 320</i>"
    }
  ]
}
```

Tabla 6. Código de carga de datos mediante JSON.

Ahora sólo queda cargar los datos, para ello, utilizando Java Script, se realizarán los siguientes pasos:

- Localizar la dirección donde se encuentran los datos a descargar (*Tabla 7*):

```
var http_request = new XMLHttpRequest();
var url = "http://url";
```

Tabla 7. Código para localizar dirección de descarga mediante JSON.

- Descargar los datos (*Tabla 8*):

```
http_request.onreadystatechange = handle_json;
http_request.open("GET", url, true);
http_request.send(null);

function handle_json() {
  if (http_request.readyState == 4) {
    if (http_request.status == 200) {
      var json_data = http_request.responseText;
      var the_object = eval("(" + json_data + ")");
    } else {
      alert("URL no válida.");
    }
    http_request = null;
  }
}
```

Tabla 8. Código para descargar datos mediante JSON.

El propio código JavaScript que Simile Timeline proporciona se encarga de interpretar el documento descargado y pintar elemento a elemento en la línea temporal.

Timemap

Timemap es una tecnología que combina una línea temporal y un mapa para representar elementos. Esta tecnología tiene en cuenta la dimensión espacial al utilizar Google Maps y la dimensión temporal al utilizar Simile Timeline.

Google Maps API y Simile Timeline API son dos APIs independientes. Dado que son dos APIs distintas se necesitan elementos intermedios que hagan de enlace, carguen los datos de una fuente en Google Maps y Simile Timeline, y muestren u oculten los distintos eventos que van sucediendo en la línea temporal. Para ello, se necesita una API llamada TimeMap API¹⁰. TimeMap ayuda a utilizar Google Maps con una línea temporal Simile Timeline. Esta librería permite cargar uno o más conjuntos de datos utilizando JSON¹¹, KML¹² o GeoRSS¹³ (entre otros) de manera simultánea en Google Maps y Simile Timeline. Permite mostrar en Google Maps todos los elementos o sólo los que aparecen en el rango visible de la línea temporal. Por defecto sólo muestra los eventos que aparecen en el rango visible de la línea temporal. Los elementos más importantes que Time Map contiene son:

- timemap.js: carga y muestra datos locales JSON.
- kml.js: cargador de archivos KML.
- flickr.js: cargador de geo-etiquetas Flickr¹⁴.
- georss.js: cargador de archivos GeoRSS.
- google_spreadsheet.js: cargador para Google Spreadsheets¹⁵ API.
- json.js: cargadores para JSON (para cadenas y jsonp).
- Conjunto de ejemplos para cada tipo de fuente de datos.
- Documentación para todos los tipos de carga de datos.

Son muchos los parámetros que se pueden modificar para personalizar un timemap. Entre otros, estos son los principales parámetros:

- Anchura y altura de la línea temporal Simile Timeline.
- Anchura y altura del mapa Google Maps.
- Iconos de Simile Timeline.
- Distintas barras de tiempo dentro de la línea temporal Simile Timeline (día, semana, año, etc.).
- Tamaño de las distintas barras de tiempo en porcentaje.
- Texto de eventos en las distintas barras de tiempo.
- Anchura de cada fracción de tiempo de cada barra de tiempo.

¹⁰ Librería para Google Maps y Simile Timeline. <http://code.google.com/p/timemap/>

¹¹ Formato para intercambio de datos. <http://es.wikipedia.org/wiki/JSON>

¹² Lenguaje de marcado basado en XML. <http://es.wikipedia.org/wiki/KML>

¹³ Estándares para representar información geográfica. <http://es.wikipedia.org/wiki/GeoRSS>

¹⁴ Definición de Flickr. <http://es.wikipedia.org/wiki/Flickr>

¹⁵ Definición de Google Docs. http://es.wikipedia.org/wiki/Google_Docs

A parte, la manera de cargar datos en Simile Timeline y Google Maps resulta de distinta forma que hacerlo en TimeMap, aunque puede ser parecida a la utilizada en Simile Timeline. En este caso, también mediante Java Script, se utilizará la clase TimeMap de TimeMap API. Esta clase se encarga de repartir los datos entre la línea temporal y el mapa. En la *Tabla 9* se puede apreciar lo fácil que resulta cargar los datos:

```
datasets: [{  
  title: "Mapas",  
  theme: "yellow",  
  type: "kml", // Data to be loaded in KML - must be a local URL  
  options: {  
    url: "xml/xml_general.kml", // KML file to load  
    "theme": customTheme,  
    "icon": customIcon  
  }  
}]
```

Tabla 9. Código de carga de datos de un TimeMap.

Se puede observar como los datos han sido cargados mediante un archivo KML que previamente ha tenido que ser generado.

A continuación se muestra un ejemplo de uso de TimeMap, en el que la parte superior corresponde a la línea temporal y la parte inferior al mapa. Este TimeMap corresponde a

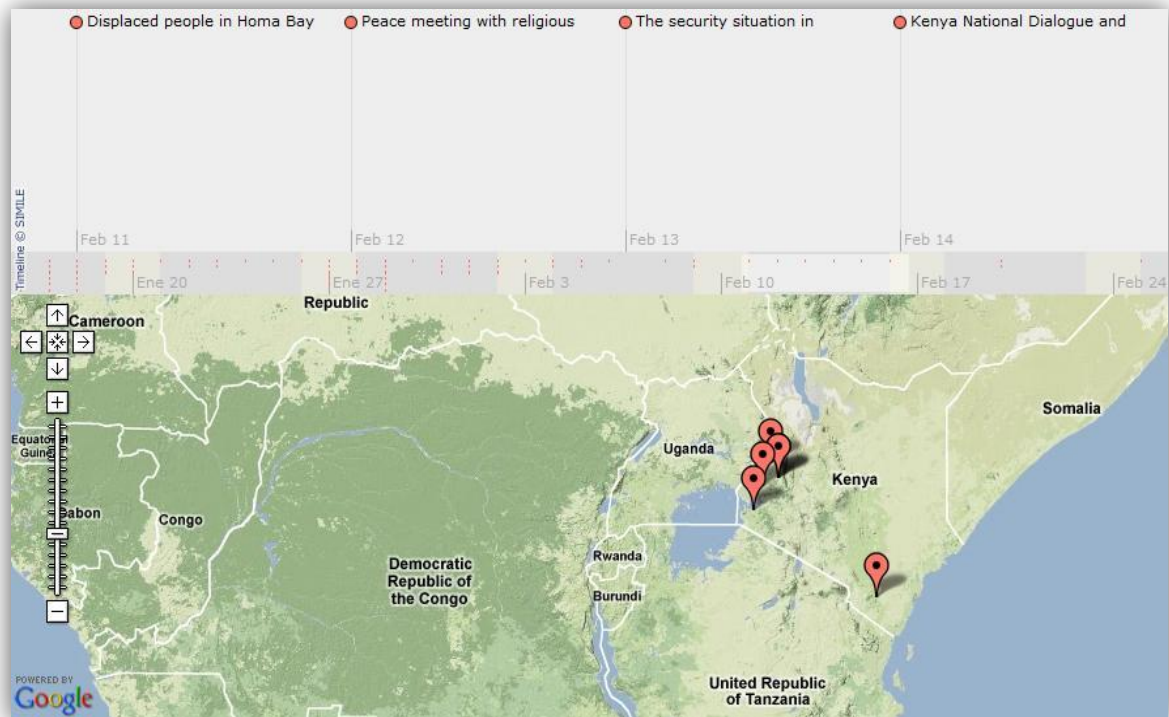


Figura 10. Ejemplo TimeMap - Ushahidi.

En la línea temporal se representan una serie de iconos rojos que representan una incidencia que tienen su correspondencia en el mapa inferior, consiguiendo saber dónde y cuándo ha ocurrido la incidencia de forma rápida. La línea temporal limita las incidencias que aparecerán,

mostrando solamente las que se hayan dado en el intervalo de tiempo que la línea temporal muestra.

3 Gestión de proyecto software

Este capítulo del documento se centra en estudiar las necesidades planteadas por los interesados en el desarrollo del producto, tratando las posibles restricciones identificadas en cualquier ámbito (legal, tecnológico, social...) que puedan condicionar el estudio. Como complemento indispensable a este análisis, se realizarán las correspondientes estimaciones en tiempo, recursos y costes. Esto ofrecerá además como resultado una descripción formal del equipo de trabajo que llevará a cabo el proyecto, así como su encuadre dentro de un plan de trabajo.

3.1 Alcance del proyecto

Definición del proyecto

Se pretende desarrollar una aplicación web que tiene como principales funciones las siguientes:

- Dotar de información espacio/temporal visual a un conjunto de elementos.
- Separar la información objetiva de la información subjetiva asociada a cada elemento.
- Posibilidad de comparar de forma visual dos conjuntos de datos utilizando su información espacio/temporal.

Estimación de tareas y recursos

En este apartado se especificarán los recursos y tareas necesarios para poder desarrollar el proyecto.

Las tareas necesarias que se deberán llevar a cabo teniendo en cuenta la metodología **Métrica V3** [9], son las siguientes:

- **EVS (estudio de viabilidad):** su objetivo es recoger los requisitos que los clientes proporcionan o que han sido adquiridos mediante entrevistas u otras técnicas.
- **ASI-DAS (análisis del sistema de información):** el objetivo es obtener una especificación detallada del sistema de información que satisfaga las necesidades de los usuarios y sirva de base para el posterior diseño del sistema.
- **DSI (diseño del sistema de información):** su propósito es realizar la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.
- **CSI (construcción del sistema de información):** esta tarea se encarga de generar el código de los componentes del sistema de información. Se desarrollan todos los procedimientos de operación y seguridad y se elaboran todos los manuales de usuario final y de explotación con el objetivo de asegurar el correcto funcionamiento del sistema para su posterior implantación.
- **IAS (implantación y aceptación del sistema):** su objetivo es la entrega y aceptación del sistema en su totalidad, y la realización de todas las actividades necesarias para que éste pase a producción.

En cuanto a los recursos que se tendrán que utilizar se mencionan los humanos, tecnológicos y físicos. Como recursos humanos es imprescindible contar con un equipo de trabajo experimentado

en el campo, que sean capaces de entender el dominio del problema y que se encarguen de proponer soluciones que cumplan los requerimientos del cliente. Como recursos físicos se necesitará material de oficina y equipos informáticos en los cuales se realizará el desarrollo del proyecto:

- **Recursos hardware:** ordenador portátil, Toshiba A-300 Intel Core 2 Dúo 2,26GHz. 4GB de memoria RAM DDR3, 500 GB HD.
- **Recursos software:**
 - Sistema operativo Windows 7
 - Java EE 1.6
 - Base de datos MySQL 5.5
 - Servidor Apache Tomcat 7.0
 - Microsoft Office 2010 Professional
 - Altova UModel 2011
 - Google Maps API 2.0
 - Simile Timeline API 1.6
 - Case Studio 2
- **Material de oficina:** folios, bolígrafos, etc.
- **Otros recursos:** conexión a internet 6MB/1MB.

Presupuesto

Primero se indicará la tarifa por hora de cada integrante del equipo y la estimación de salario bruto mensual, calculado sobre un promedio de 8 horas diarias y 22 días al mes.

Cargo	Euros/Hora	Horas/Día	Días/Mes	Bruto mensual
Jefe de proyecto	27,40	8	22	4.822,40
Analista	12,45	8	22	2.191,20
Arquitecto	12,45	8	22	2.191,20
Diseñador	12,45	8	22	2.191,20
Gestión de configuración	19,30	8	22	3.396,80
Gestión de calidad	19,30	8	22	3.396,80
Gestión del proyecto	19,30	8	22	3.396,80
			Total:	21.586,40 €

Tabla 10. Salario bruto mensual.

A continuación se establecen los costes de equipos y dispositivos necesarios para el desarrollo e implantación del sistema:

Descripción	Coste/Unidad	Unidades	Coste Total
Portátil Toshiba A-300	710,10	1	710,10
Servidor HP ProLiant ML115 G5	823,34	1	823,34
Switch 16 puertos 10/100 Ovislink 19	56,67	1	56,67
Ratón Logitech Óptico M115	12,73	1	12,73
		Total:	1.602,84 €

Tabla 11. Coste de equipos y dispositivos.

En la siguiente tabla se indican los costes relacionados con el software que se va a utilizar.

Descripción	Coste/Unidad	Unidades	Coste Total
-------------	--------------	----------	-------------

Windows 7 Professional	309,00	1	309,00
Microsoft Office 2010	699,00	1	699,00
Altova UModel 2011 Enterprise Edition	199,00	1	199,00
MySQL 5.5 [10]	0	1	0
Apache Tomcat 7.0 [11]	0	1	0
Case Studio 2	435,00	1	435,00
Eclipse IDE [12]	0	1	0
		Total:	1.642,00 €

Tabla 12. Coste de software.

A continuación se indican los costes relacionados con internet y hosting.

Descripción	Coste/mes	Coste/año	Coste Total
Hosting servidor	8,00	96,00	96,00
Consumo ancho de bando: 10GB/mes	7,00	84,00	84,00
Dominio	1,00	12,00	12,00
		Total:	192,00 €

Tabla 13. Costes de hosting y conectividad.

En la siguiente tabla se presentan los gastos en material de oficina.

Descripción	Coste/unidad	Unidades	Coste Total
Material de papelería	N/A	N/A	200,00
		Total:	200,00 €

Tabla 14. Costes de material de oficina.

A continuación se especifica el coste total de implementación e instalación del producto.

Descripción	Coste/mes	Nº meses	Coste Total
Implementación	21.586,40	3	64.759,20
Instalación	2.500,00	1	2.500,00
		Total:	67.259,20 €

Tabla 15. Costes de implementación e instalación del producto.

Por último se especifica el coste total del proyecto realizando la suma de cada gasta descrito anteriormente.

Descripción	Coste	Impuesto	Total impuestos	Total
Implementación	64.759,20	-	-	64.759,20
Instalación	2.500,00	-	-	2.500,00
Hosting y conectividad	192,00	18 %	34,56	226,56
Software	1.642,00	18 %	295,56	1.937,56
Equipos y dispositivos	1.602,84	18 %	288,51	1.892,35
Material de oficina	200,00	18 %	36,00	236,00
Total:			576,33 €	71.556,02 €

Tabla 16. Costes totales.

Teniendo en cuenta los riesgos que pueden ocurrir, la empresa debe ser capaz de afrontarlos sin tener que cobrar más al cliente, por eso se incluye un porcentaje del 15% para imprevistos. Por

último, hay que incluir un margen de beneficios para la empresa que dado la duración y dificultad de la aplicación será de un 20%.

Descripción	Coste
Coste total	71.556,02 €
Beneficio	14.207,67 €
Imprevistos	10.655,76 €
Total	96.419,45 €

Tabla 17. Cálculo total.

3.2 Plan de trabajo

En este apartado se detallan todas las tareas necesarias que se tienen que llevar a cabo para poder desarrollar el sistema con un mínimo de calidad, gestión y orden. Para ello se utilizará la metodología Métrica V3.

Identificación de tareas

Todas las tareas han sido identificadas en la metodología Métrica V3.

La primera tarea corresponde con la **Gestión del Proyecto**. En esta tarea se realiza el *Documento de Oferta*, *Documento de Cálculo de Costes*, *Documento de Plan de Gestión de la Configuración* y el *Documento de Plan de Gestión de Calidad*.

La segunda tarea corresponde con el **Estudio de Viabilidad del Sistema**. Esta tarea se encarga de escribir los requisitos del sistema, proponer varias opciones y seleccionar una de ellas.

La siguiente tarea es **Análisis del Sistema**. En esta etapa se describen de forma detallada todos los requisitos del sistema, se identifican los subsistemas de análisis, se desarrollan los casos de uso y se realiza un análisis previo de las clases. Además se realiza una elaboración del modelo de datos, procesos e interfaces de usuario previas a la siguiente tarea, dónde se desarrollará en profundidad. Por último se especifica un plan de pruebas tomando como base los requisitos descritos.

La cuarta tarea corresponde con el **Diseño del Sistema**. En esta tarea se especifica la estructura que tendrá la aplicación a desarrollar. Se define la arquitectura del sistema, arquitectura de soporte, casos de uso reales, diseño de clases, arquitectura de módulos del sistema, diseño físico de datos, planes de prueba y requisitos de implantación.

La quinta tarea es **Construcción del Sistema de Información**. En esta etapa de desarrollo se genera el código del sistema, se ejecutan las pruebas unitarias, de integración, de sistema y de aceptación. Por último se desarrolla el manual de usuario, se forma a los usuarios y se aprueba el sistema construido.

La última etapa es **Plan de Implantación del Sistema**. En esta última etapa se pone en funcionamiento el sistema construido y se pasa a presentar y aprobar el sistema. Si el sistema es aceptado se pasa a producción.

Estimación de tareas

En este apartado se determinará el tiempo que se empleará en cada tarea teniendo en cuenta los objetivos que se quieren alcanzar y los recursos disponibles. Los recursos temporales son de 3 meses, por lo que el cómputo total de desarrollo del proyecto no debe ser superior. A

continuación se muestra una tabla que contiene todas las tareas, fecha de inicio, duración y fecha de finalización.

Actividades	Fecha de inicio	Duración (días)	Fecha de fin
Gestión del Proyecto	30/08/2010	14	12/09/2010
Estudio de Viabilidad del Sistema	13/09/2010	7	19/09/2010
Análisis del Sistema	20/09/2010	14	03/10/2010
Diseño del Sistema	04/10/2010	14	17/10/2010
Construcción del Sistema de Información	18/10/2010	21	07/11/2010
Ejecución de las Pruebas Unitarias	08/11/2010	7	14/11/2010
Plan de Implantación del Sistema	15/11/2010	14	28/11/2010
Formación necesaria para la Implantación	26/11/2010	2	28/11/2010
Formación del equipo de Implantación	26/11/2010	2	28/11/2010
Formación de Administradores	26/11/2010	2	28/11/2010

Tabla 18. Estimación de tareas.

Planificación de tareas

Para representar la planificación se ha decidido utilizar el Diagrama de Gantt. En él se pueden observar todas las tareas, su duración y si éstas se solapan con otras tareas.

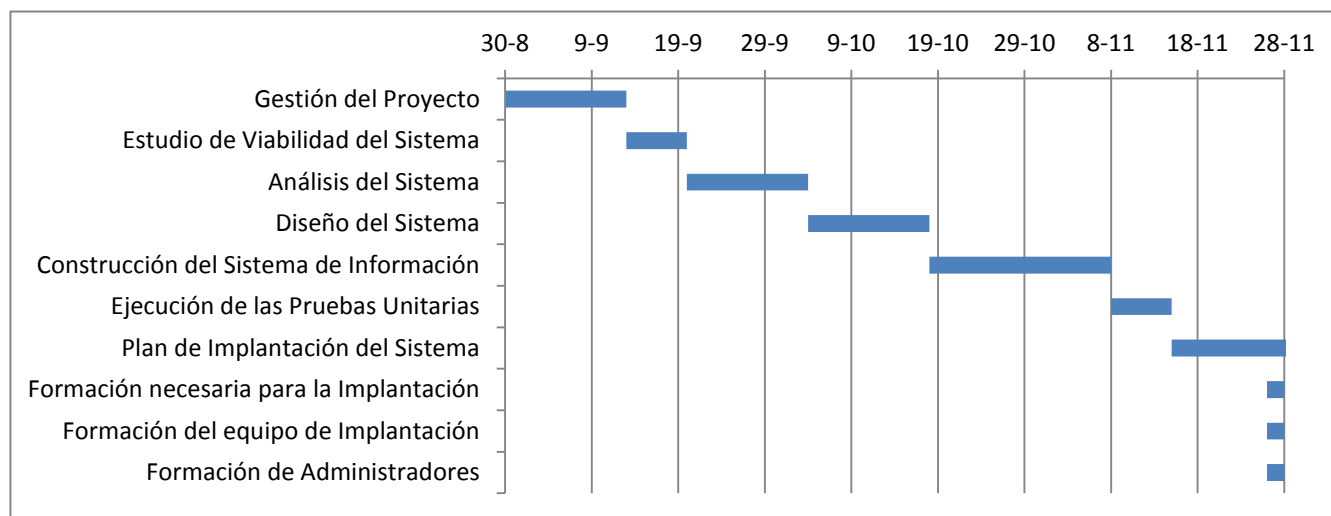


Figura 11. Diagrama de Gantt.

3.3 Gestión de recursos

Especificación de recursos

El equipo de trabajo estará formado por personas especializadas en el dominio del desarrollo del software y existirán los siguientes cargos:

- **Jefe de proyecto:** encargado de coordinar, dirigir, controlar e integrar al resto de las personas participantes en el grupo de trabajo.

- **Comité de calidad:** colaboradores del departamento de calidad. Su función es garantizar la calidad de los distintos productos. De esta manera el cliente puede confiar en que los productos están satisfaciendo sus metas.
- **Comité de gestión de configuración:** colaboradores del departamento de calidad y su función es realizar toda la documentación necesaria sobre los cambios producidos en los diferentes productos del proyecto. Es decir, realizan diferentes versiones del documento.
- **Comité de gestión del proyecto:** colaboradores del departamento de gestión del proyecto. Se encargan de cumplir plazos de entregas.
- **Responsables de comité:** máximos responsables del comité de calidad, gestión de configuración y gestión del proyecto.
- **Analista:** responsable de investigar, planear, coordinar y recomendar opciones de software y sistemas para cumplir con los requerimientos de la empresa.
- **Arquitecto:** encargado de realizar el diseño de más alto nivel de la estructura del sistema.
- **Diseñador:** encargado de diseñar posibles soluciones al problema que se plantea, bajo la supervisión del jefe de proyecto.



Figura 12. Organigrama.

Asignación de recursos

Los recursos, tanto humanos como tangibles, se asignarán dependiendo de la fase del proyecto en la que se esté (Figura 11. Diagrama de Gantt.).

3.4 Gestión de riesgos

En esta sección se va a abordar el plan que se tiene que llevar a cabo ante la aparición de riesgos típicos en el desarrollo de una aplicación informática, con el propósito de poder evitarlos, o

bien solucionarlos de la forma más eficiente posible. Se identificarán los riesgos potenciales, se valorará su probabilidad y su impacto y se establecerá una prioridad según su importancia. Después se establecerá un plan para controlar los riesgos.

Identificación de riesgos

El desarrollo de sistemas de software es un proceso costoso, difícil y con altos índices de fracaso. Siempre está presente el riesgo de que ciertos eventos se traduzcan en complicaciones para el desarrollo del proyecto. Para intentar evitar o minimizar estas complicaciones se tratarán de identificar los posibles riesgos.

Primero se deben identificar las fuentes de riesgos para que los propios riesgos sean más fáciles de gestionar:

Fuente	Descripción
Naturales	Son aquellas complicaciones que pueden aparecer como consecuencia de catástrofes naturales, como terremotos, inundaciones, incendios, etc.
Internas	Son los riesgos relacionados con el desarrollo del sistema de información y con los recursos de la empresa, sin incluir al personal.
Externas	Se corresponden con las dificultades que no pertenecen a la empresa, es decir, son ajenas a ella y tampoco son naturales.
Personal	Estos riesgos están relacionados con la documentación del equipo de desarrollo con el cliente o bien con asuntos personales de los propios miembros del equipo de desarrollo.

Tabla 19. Fuentes de riesgos.

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos, etc.). Al identificar los riesgos que pueden darse resulta posible poder controlarlos. A continuación se listan y detallan cada uno de los riesgos que se han identificado en este momento del ciclo de vida del sistema de información.

Riesgos naturales

RIESGO-01			
Nombre:	Terremoto		
Fuente:	Naturales		
Probabilidad de ocurrencia:	1%	Impacto:	Muy alto
Descripción	Movimiento de tierra que produce un destrozo en las infraestructuras y en el edificio de trabajo.		
Consecuencias:	Destrucción total o parcial de los equipos de trabajo, Además se produciría la pérdida de todo el material que está almacenado dentro de los equipos afectados.		

Tabla 20. Riesgo Terremoto.

RIESGO-02			
Nombre:	Fallo eléctrico		
Fuente:	Naturales		
Probabilidad de ocurrencia:	9%	Impacto:	Muy alto
Descripción	Subida de tensión, o bien falta de corriente eléctrica que imposibilita el uso de los equipo informáticos. Este riesgo puede producirse por una tormenta eléctrica o por fallos en el suministro eléctrico.		
Consecuencias:	Destrucción total o parcial de los equipos de trabajo, Además se produciría la pérdida de todo el material que está almacenado dentro de los equipos afectados.		

Tabla 21. Riesgo Eléctrico.

RIESGO-03			
Nombre:	Incendio		
Fuente:	Naturales		
Probabilidad de ocurrencia:	8%	Impacto:	Medio
Descripción	Fuego provocado por cortocircuitos, subidas de tensión, cigarrillos alojados en las papeleras. Los incendios pueden ser provocados.		
Consecuencias:	Destrucción total o parcial de los equipos de trabajo que se encuentran en el foco del incendio, En ocasiones graves, se puede producir la pérdida de empleados.		

Tabla 22. Riesgo Incendio.

RIESGO-04			
Nombre:	Inundación		
Fuente:	Naturales		
Probabilidad de ocurrencia:	7%	Impacto:	Muy alto
Descripción	Exceso de agua en el puesto de trabajo ya sea por lluvias torrenciales o por filtraciones que produce un peligro en el desarrollo del trabajo.		
Consecuencias:	Destrucción total o parcial de los equipos de trabajo. Además se produciría la pérdida de todo el material que está almacenado dentro de los equipos afectado.		

Tabla 23. Riesgo Inundación.

Riesgos de personal

RIESGO-05			
Nombre:	Problemas de comunicación con el cliente		
Fuente:	Personal		
Probabilidad de ocurrencia:	85%	Impacto:	Bajo
Descripción	Falta de entendimiento entre el equipo de desarrollo y el cliente, como consecuencia de una falta de reuniones, de capacidad de expresión por parte del cliente, etc.		
Consecuencias:	Errores en la extracción de requisitos, en la documentación, retrasos en las entregas y pérdida de confianza o resignación del cliente.		

Tabla 24. Riesgo Comunicación.

RIESGO-06			
Nombre:	Expectativas erróneas sobre un miembro del equipo		
Fuente:	Personal		
Probabilidad de ocurrencia:	8%	Impacto:	Medio
Descripción	Existe un miembro que no cumple con los requisitos mínimos de trabajo que se le asigna dentro del equipo de desarrollo.		
Consecuencias:	Baja productividad, provocando mayor carga en el resto de integrantes del grupo o un retraso en la planificación.		

Tabla 25. Riesgo Expectativas.

RIESGO-07			
Nombre:	Baja temporal		
Fuente:	Personal		
Probabilidad de ocurrencia:	35%	Impacto:	Bajo
Descripción	Riesgo provocado por la ausencia de algún miembro del personal como consecuencia de enfermedad, asuntos personales, accidente laboral o doméstico, embarazo, etc. En definitiva, toda falta que se produzca en un periodo de tiempo menor al 30% del tiempo de planificación estipulada en el proyecto.		
Consecuencias:	Dicha ausencia provocará que el trabajo que este miembro del equipo realizaba, tenga que ser suplido por otro miembro del equipo. Es decir, mayor carga de trabajo para el resto de miembros del equipo.		

Tabla 26. Riesgo Baja Temporal.

RIESGO-08			
Nombre:	Sabotaje interno		
Fuente:	Personal		
Probabilidad de ocurrencia:	1%	Impacto:	Alto
Descripción	Perjuicio del proyecto producido por un miembro del equipo de desarrollo.		
Consecuencias:	Destrucción de los datos que tiene acceso el integrante del equipo.		

Tabla 27. Riesgo Sabotaje Interno.

RIESGO-09			
Nombre:	Baja permanente		
Fuente:	Personal		
Probabilidad de ocurrencia:	5%	Impacto:	Alto
Descripción	Cuando un miembro del grupo, que no se corresponde con un responsable de comité o Jefe de proyecto, es despedido; resigna o sufre un suceso que le impide trabajar con el resto del equipo de desarrollo. En definitiva toda falta que se produzca en un periodo de tiempo mayor al 30% del tiempo de planificación estipulada en el proyecto.		
Consecuencias:	Dicha ausencia provocará que el trabajo que este miembro del equipo realizaba, tenga que ser suplido por otro miembro del equipo. Es decir, mayor carga de trabajo para el resto de miembros del equipo.		

Tabla 28. Riesgo Baja Permanente.

RIESGO-10			
Nombre:	Baja del Jefe de Proyecto		
Fuente:	Personal		
Probabilidad de ocurrencia:	15%	Impacto:	Alto
Descripción	Cuando el Jefe de proyecto, es despedido; resigna o sufre un suceso que le impide trabajar con el resto del equipo de desarrollo o abandona su cargo. En definitiva toda falta que se produzca en un periodo de tiempo mayor al 30% del tiempo de planificación estipulada en el proyecto.		
Consecuencias:	Dicha ausencia, debida a la importancia del cargo, deberá ser suplida por otro miembro del equipo de desarrollo y nombrarle nuevo Jefe de Proyecto.		

Tabla 29. Riesgo Baja del Jefe de Proyecto.

RIESGO-11			
Nombre:	Baja de un responsable de comité		
Fuente:	Personal		
Probabilidad de ocurrencia:	20%	Impacto:	Medio
Descripción	Cuando un responsable de alguno de los comités es despedido o sufre un suceso que le impide trabajar con el resto del equipo de desarrollo. En definitiva, toda falta que se produzca en un periodo de tiempo mayor al 30% del tiempo de planificación estipulada en el proyecto.		
Consecuencias:	Dicha ausencia, debida a la importancia del cargo, deberá ser suplida por el subordinado en el comité en cuestión y nombrarle nuevo Responsable del Comité.		

Tabla 30. Riesgo Baja de algún Responsable de Comité.

Riesgos externos

RIESGO-12			
Nombre:	Modificación de requisitos		
Fuente:	Externos		
Probabilidad de ocurrencia:	68%	Impacto:	Medio
Descripción	Cambio en las exigencias del cliente, o bien en el enfoque del desarrollo del proyecto.		
Consecuencias:	Dependiendo del nivel de modificación de las especificaciones del software, las consecuencias serán más profundas o menos profundas.		

Tabla 31. Riesgo Modificación de Requisitos.

RIESGO-13			
Nombre:	Sabotaje externo		
Fuente:	Externos		
Probabilidad de ocurrencia:	3%	Impacto:	Medio
Descripción	Ataque al proyecto producido por una persona ajena al equipo de desarrollo.		
Consecuencias:	Destrucción o modificación de los datos a los que se ha tenido acceso.		

Tabla 32. Riesgo Sabotaje Externo.

RIESGO-14			
Nombre:	Cancelación del proyecto		
Fuente:	Externos		
Probabilidad de ocurrencia:	1%	Impacto:	Muy alto
Descripción	El cliente decide retirar su deseo de que se desarrolle el proyecto.		
Consecuencias:	Pérdida de tiempo, dinero, recursos, etc.		

Tabla 33. Riesgo Cancelación del Proyecto.

RIESGO-15			
Nombre:	Espionaje industrial		
Fuente:	Externos		
Probabilidad de ocurrencia:	25%	Impacto:	Medio
Descripción	Robo o compra de información por parte de la competencia.		
Consecuencias:	Pérdida de confianza por parte del cliente. En el peor de los casos se puede cancelar el proyecto puesto que la competencia realiza una mejor oferta al cliente.		

Tabla 34. Riesgo Espionaje.

Riesgos internos

RIESGO-16			
Nombre:	Error en los requisitos		
Fuente:	Internos		
Probabilidad de ocurrencia:	20%	Impacto:	Muy alto
Descripción	Error grave en un requisito (u omisión del mismo).		
Consecuencias:	Rehacer todos los documentos relacionados con ese documento. La diferencia con el RIESGO-10 es que este riesgo se produce por un fallo de un miembro del equipo de desarrollo, mientras que el otro, es un deseo del cliente.		

Tabla 35. Riesgo Error en Requisito.

RIESGO-17			
Nombre:	Saturación de recursos		
Fuente:	Internos		
Probabilidad de ocurrencia:	12%	Impacto:	Medio
Descripción	Intentar realizar más trabajo del que se puede realizar con los recursos disponibles.		
Consecuencias:	Retrasos en la entrega de documentos, y creación de una sensación de malestar por parte del cliente. Además se produce un retraso progresivo en las entregas, así como una concentración de carga de trabajo en las fases posteriores.		

Tabla 36. Riesgo Saturación de Recursos.

RIESGO-18			
Nombre:	Desviación/Error en la planificación		
Fuente:	Internos		
Probabilidad de ocurrencia:	30%	Impacto:	Alto
Descripción	La asignación de trabajo y distribución del mismo para realizar la fase correspondiente es errónea. También puede aparecer como consecuencia de una desviación en la planificación.		
Consecuencias:	Retrasos en la entrega de documentos, y creación de una sensación de malestar por parte del cliente. Además se produce un retraso progresivo en las entregas, así como una concentración de carga de trabajo en las fases posteriores.		

Tabla 37. Riesgo Desviación/Error en la planificación.

RIESGO-19			
Nombre:	Incompatibilidad de versiones		
Fuente:	Internos		
Probabilidad de ocurrencia:	55%	Impacto:	Bajo
Descripción	Las versiones no coinciden con la revisión establecida, o se trabaja sobre una versión anulada o caduca.		
Consecuencias:	Pérdida de información crítica. Creación de incertidumbre sobre el documento final.		

Tabla 38. Riesgo Incompatibilidad de Versiones.

RIESGO-20			
Nombre:	Fallos de documentación		
Fuente:	Internos		
Probabilidad de ocurrencia:	95%	Impacto:	Medio
Descripción	Errores detectados por el cliente y/o el equipo de desarrollo en el documento a revisar.		
Consecuencias:	Reescribir las partes que no son válidas o desechadas.		

Tabla 39. Riesgo Fallos de Documentación.

RIESGO-21			
Nombre:	Malestar en el seno del equipo		
Fuente:	Internos		
Probabilidad de ocurrencia:	15%	Impacto:	Medio
Descripción	Deterioro en el ambiente de trabajo del equipo.		
Consecuencias:	Disminución de la productividad de los miembros del equipo afectados, falta de reuniones, mala comunicación, etc.		

Tabla 40. Riesgo Malestar en el Seno del Equipo.

RIESGO-22			
Nombre:	Destrucción de las infraestructuras de trabajo		
Fuente:	Internos		
Probabilidad de ocurrencia:	2%	Impacto:	Muy alto
Descripción	Pérdida de material informático o recursos no personales.		
Consecuencias:	Pérdida de todos los datos asociados a ese material, además aumento del coste como consecuencia de la compra de los nuevos materiales.		

Tabla 41. Riesgo Destrucción de las Infraestructuras de Trabajo.

RIESGO-23			
Nombre:	Retrasos en la entrega		
Fuente:	Internos		
Probabilidad de ocurrencia:	4%	Impacto:	Alto
Descripción	Entregar documentos fuera de la fecha de entrega del mismo.		
Consecuencias:	Pérdida de confianza por parte del cliente, concentración de carga de trabajo en las fases posteriores y aumento del riesgo de cancelación del proyecto.		

Tabla 42. Riesgo Retrasos en la Entrega.

Análisis de riesgos

Una vez documentados y descritos los riesgos, se va a analizar cada uno de ellos. Este análisis consistirá en un análisis cualitativo y cuantitativo.

Análisis cualitativo

En esta sección se estima el impacto y la probabilidad que tienen cada riesgo identificado. Si el riesgo analizado requiere una respuesta inmediata para evitar sus consecuencias, se incrementa la importancia de dicho riesgo, pues requiere una atención especial. El análisis, puesto que es cualitativo, se realiza utilizando términos como *'muy alto'*, *'alto'*, etc., para las consecuencias de su aparición.

A continuación se muestra el valor que puede tener el impacto de un riesgo sobre la planificación, el coste, el alcance y la calidad final del producto.

Objetivos	Muy bajo	Bajo	Moderado	Alto	Muy alto
Coste	Despreciable incremento de coste	<3.500€ incremento de coste	3.500 – 6.500 € incremento de coste	6.500 – 12.500 € incremento de coste	> 12.500 € incremento de coste
Calendario	Retraso insignificante	Rehacer varias tareas < 12 días	Re planificación de una fase 12-25 días	Re planificación de varias fases 25-50 días	Realizar una nueva planificación > 50 días
Alcance	Cambios insignificantes	Modificación de la versión del documento	Rechazo de uno de los documento	Rechazo de varios de los documentos	Los productos finales son inútiles
Calidad	Degradación de la calidad insignificante	Solo algunas aplicaciones están afectadas	La reducción de calidad requiere aprobación del cliente	La reducción de calidad es inaceptable por el cliente	El producto final es inútil

Tabla 43. Impacto de in riesgo en el desarrollo.

A continuación se muestra la en la que se evalúa cada riesgo con los parámetros anteriores.

Riesgo	Coste	Calendario	Alcance	Calidad
RIESGO-01	Muy alto	Alto	Muy alto	Muy alto
RIESGO-02	Muy alto	Alto	Muy alto	Muy alto
RIESGO-03	Alto	Alto	Bajo	Bajo
RIESGO-04	Alto	Alto	Alto	Moderado
RIESGO-05	Muy bajo	Bajo	Bajo	Bajo
RIESGO-06	Muy bajo	Alto	Muy bajo	Moderado
RIESGO-07	Muy bajo	Moderado	Muy bajo	Moderado
RIESGO-08	Alto	Alto	Alto	Alto
RIESGO-09	Bajo	Muy alto	Bajo	Alto
RIESGO-10	Moderado	Moderado	Alto	Alto
RIESGO-11	Bajo	Moderado	Moderado	Moderado
RIESGO-12	Moderado	Moderado	Alto	Bajo
RIESGO-13	Alto	Alto	Bajo	Bajo
RIESGO-14	Muy alto	Muy alto	Muy alto	Muy alto
RIESGO-15	Muy bajo	Muy bajo	Muy alto	Muy alto

RIESGO-16	Alto	Alto	Alto	Muy alto
RIESGO-17	Moderado	Alto	Moderado	Moderado
RIESGO-18	Alto	Alto	Moderado	Bajo
RIESGO-19	Bajo	Bajo	Alto	Alto
RIESGO-20	Bajo	Moderado	Moderado	Moderado
RIESGO-21	Bajo	Moderado	Moderado	Moderado
RIESGO-22	Muy alto	Muy alto	Muy alto	Muy alto
RIESGO-23	Alto	Alto	Moderado	Alto

Tabla 44. Análisis cualitativo de los riesgos.

Análisis cuantitativo

El análisis cuantitativo pretende calcular el factor de riesgo (FR). Con el cálculo de esta variable, se va a poder cuantificar de forma numérica el riesgo en base a la probabilidad de aparición sobre el desarrollo. De esta forma se va a poder dar prioridad a cada riesgo en función de este valor, y su cálculo se realiza con la siguiente fórmula:

$$FR = Valor\ ocurrencia \cdot Impacto$$

Se ha establecido la relación entre el nivel de impacto y su valor numérico de la siguiente manera:

Nivel de impacto	Valor numérico
Muy bajo	0,1
Bajo	0,3
Moderado	0,5
Alto	0,7
Muy alto	0,9

Tabla 45. Relación numérica del impacto.

Cuanto mayor sea FR, mayor es la probabilidad de que aparezca el riesgo en cuestión y su impacto, y por lo tanto, dicho riesgo es más importante de evitar para que no afecte al desarrollo del sistema. A continuación se muestran todos los FR de los riesgos que se han detectado:

Riesgo	Valor ocurrencia	Impacto	FR
RIESGO-01	0,01	0,9	0,009
RIESGO-02	0,09	0,9	0,027
RIESGO-03	0,08	0,5	0,072
RIESGO-04	0,07	0,9	0,063
RIESGO-05	0,85	0,3	0,255
RIESGO-06	0,08	0,5	0,040
RIESGO-07	0,35	0,3	0,105
RIESGO-08	0,01	0,7	0,007
RIESGO-09	0,05	0,7	0,035
RIESGO-10	0,15	0,7	0,105
RIESGO-11	0,20	0,5	0,100
RIESGO-12	0,68	0,5	0,340
RIESGO-13	0,03	0,5	0,015
RIESGO-14	0,01	0,9	0,009
RIESGO-15	0,25	0,5	0,125

RIESGO-16	0,20	0,9	0,180
RIESGO-17	0,12	0,5	0,060
RIESGO-18	0,30	0,7	0,210
RIESGO-19	0,55	0,3	0,165
RIESGO-20	0,95	0,5	0,475
RIESGO-21	0,15	0,5	0,075
RIESGO-22	0,02	0,9	0,018
RIESGO-23	0,04	0,7	0,028

Tabla 46. Factores de Riesgo individuales.

La suma total de todos los factores de riesgo es 2,313, y este valor se utiliza para calcular el FR medio de un riesgo. Para ello se parte de la fórmula del factor de riesgo, pero en vez de trabajar con los valores de ocurrencias e impactos propios de cada uno, se deben calcular los valores medios:

	Valor numérico
Valor ocurrencia medio	0,25
Impacto medio	0,67
FR medio	0,1675

Tabla 47. Cálculo del factor de riesgo medio.

Con esta medida, el FR medio, se puede calcular la media de influencia que un riesgo tiene sobre el desarrollo del software, relacionando su probabilidad de aparición media y su impacto medio sobre el desarrollo del software. Trabajando con probabilidades, se obtiene que un riesgo afecte en un 16,75% sobre el desarrollo software, si por algún motivo apareciese. Esto quiere decir que de media, si un riesgo aparece influye aproximadamente un 16,75% del proyecto.

3.5 Plan de pruebas

En este apartado se presenta el alcance de las pruebas a las que será sometida la aplicación una vez haya sido implementado. También se definirá la estrategia a seguir, los objetivos de las pruebas, los recursos que se tienen que utilizar y su ubicación en las tareas definidas.

Definición de pruebas (alcance pruebas y estrategia (tipos pruebas) a seguir)

El propósito del plan de pruebas es desarrollar un conjunto de pruebas que validen el sistema desarrollado, es decir, verificar que cumple con todos los requisitos descritos. Al ejecutar todas las pruebas se sabrá si su diseño es correcto antes de empezar la fase de implantación y paso a producción del sistema.

La estrategia a seguir es definir un conjunto de pruebas unitarias para cada subsistema (Figura 14. Diagrama de subsistemas.) detectadas en la fase de diseño, con el objeto de comprobar que su estructura es correcta y que se ajustan a la funcionalidad establecida.

Las pruebas que se definirán son:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas de implantación.
- Pruebas de aceptación.

Para cada prueba se tiene que definir el hardware mínimo necesario para ejecutar la prueba, los equipos informáticos que tomarán parte y el plan de ejecución que hay que seguir.

Las pruebas unitarias tienen dos conjuntos de pruebas bien diferenciados:

- Pruebas de caja blanca: se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Se escogerán distintos valores de entrada para cada procedimiento y cerciorarse de que se devuelven los valores de salida adecuados.
- Pruebas de caja negra: se centra en pruebas en las que no se tiene en cuenta el código, sino la respuesta a una serie de entradas. En otras palabras, estas pruebas se centran en comprobar la forma de interactuar con el medio teniendo en cuenta que debe hacer pero no como debe hacerlo.

Especificación de pruebas (objetivos esperados, recursos utilizar, calendario pruebas)

El objetivo de las distintas pruebas es el siguiente:

- Pruebas unitarias: validan el correcto funcionamiento de un módulo de código. Sirven para asegurar que cada uno de los módulos funciona correctamente por separado.
- Pruebas de integración: forman la fase del testeo de software en la cual módulos individuales de software son combinados y testeados como un grupo, verificando que todas las partes del software funcionan unidas.
- Pruebas de implantación: la finalidad de las pruebas de implantación es doble:
 - Comprobar el funcionamiento correcto del mismo en el entorno de operación.
 - Permitir que el usuario determine, desde el punto de vista de operación, la aceptación del sistema instalado en su entorno real, según el cumplimiento de los requisitos especificados.

- Pruebas de aceptación: tienen como fin validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir que el usuario determine la aceptación del sistema. Durante este periodo de tiempo, el usuario debe plantear todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente.

A continuación se definen los requisitos para cada prueba.

Pruebas unitarias	
Hardware mínimo	<ul style="list-style-type: none"> • Procesador Intel Core 2 Dúo 2,26 GHz. • 4GB memoria RAM. • 200GB de disco duro. • Pantalla de 15". • Tarjeta de red inalámbrica. • Tarjeta gráfica ATI.
Equipos informáticos	<ul style="list-style-type: none"> • Portátil Toshiba A-300. • Windows 7 Professional. • Servidor Apache Tomcat 7. • Base de datos MySQL 5.5. • Java EE 1.6. • Conexión a internet.
Plan de ejecución	<ul style="list-style-type: none"> • Se prepara el entorno de ejecución. • Se ejecuta la prueba unitaria. • Se estudian los resultados obtenidos.

Tabla 48. Pruebas unitarias.

Pruebas de integración	
Hardware mínimo	<ul style="list-style-type: none"> • Procesador Intel Core 2 Dúo 2,26 GHz. • 4GB memoria RAM. • 200GB de disco duro. • Pantalla de 15". • Tarjeta de red inalámbrica. • Tarjeta gráfica ATI.
Equipos informáticos	<ul style="list-style-type: none"> • Portátil Toshiba A-300. • Windows 7 Professional. • Servidor Apache Tomcat 7. • Base de datos MySQL 5.5. • Java EE 1.6. • Conexión a internet.
Plan de ejecución	<ul style="list-style-type: none"> • Se prepara el entorno de ejecución. • Se ejecuta la aplicación. • Se ejecutan las pruebas de integración. • Se estudian los resultados obtenidos.

Tabla 49. Pruebas de integración.

Pruebas de implantación	
Hardware mínimo	<ul style="list-style-type: none"> • Procesador Intel Core 2 Dúo 2,26 GHz. • 4GB memoria RAM. • 200GB de disco duro. • Pantalla de 15". • Tarjeta de red inalámbrica. • Tarjeta gráfica ATI.
Equipos informáticos	<ul style="list-style-type: none"> • Portátil Toshiba A-300. • Windows 7 Professional. • Servidor Apache Tomcat 7. • Base de datos MySQL 5.5. • Java EE 1.6. • Conexión a internet.
Plan de ejecución	<ul style="list-style-type: none"> • Se prepara el entorno de ejecución. • Se ejecuta la aplicación. • Se ejecutan las pruebas de implantación. • Se estudian los resultados obtenidos.

Tabla 50. Pruebas de implantación.

Pruebas de aceptación	
Hardware mínimo	<ul style="list-style-type: none"> • Procesador Intel Core 2 Dúo 2,26 GHz. • 4GB memoria RAM. • 200GB de disco duro. • Pantalla de 15". • Tarjeta de red inalámbrica. • Tarjeta gráfica ATI.
Equipos informáticos	<ul style="list-style-type: none"> • Portátil Toshiba A-300. • Windows 7 Professional. • Servidor Apache Tomcat 7. • Base de datos MySQL 5.5. • Java EE 1.6. • Conexión a internet.
Plan de ejecución	<ul style="list-style-type: none"> • Se prepara el entorno de ejecución. • Se ejecuta la aplicación. • Se abre un navegador web compatible. • Se carga la aplicación web en el navegador. • Se ejecutan las pruebas de aceptación. • Se estudian los resultados obtenidos.

Tabla 51. Pruebas de aceptación.

El conjunto de las pruebas unitarias deben ejecutarse una vez el sistema ha sido implementado. Las fechas de ejecución corresponden con el intervalo de 08/11/2010 al 14/11/2010. Una vez comprobadas que las pruebas unitarias devuelven los resultados esperados se realizan las pruebas de integración para comprobar que el sistema funciona correctamente al

interaccionar todos los módulos. A continuación se llevan a cabo las pruebas de implantación para comprobar que el sistema se puede desplegar.

Por último se ejecutan las pruebas de aceptación. Estas pruebas las tiene que realizar el cliente y servirá para comprobar que la aplicación cumple con todas las peticiones que el cliente transmitió o para encontrar errores y poder solucionarlos. Las pruebas de integración, implantación y aceptación se ejecutarán entre el 15/11/21010 y el 28/11/2010.

El conjunto de estas tareas está especificado en el diagrama de Gantt construido (Figura 11. Diagrama de Gantt.).

4 Solución

En este apartado se explicará la solución que ha sido elegida, aportando una serie de requisitos que se deberán cumplir para satisfacer las distintas funcionalidades. A continuación, se expondrá el diseño que respete los requisitos establecidos. Por último se realizará la implementación y el conjunto de pruebas del sistema.

4.1 Descripción de la solución

Los *mashups*¹⁶ son aplicaciones web que reúnen una gran cantidad de datos creando un nuevo contenido mucho más completo que cada fuente por separado. Los *mashups* no necesitan necesariamente almacenar los datos que vayan a consumir, simplemente hay que indicar dónde tiene que acceder y cómo tratarlos dependiendo de la petición del usuario. Entre otras funciones, los mashups pueden acceder a varias fuentes de información y representarlas dentro de un mismo campo, como por ejemplo un mapa o una línea temporal. En ellos se puede identificar cualquier información de forma rápida y acceder a datos más específicos a través de dicha información.

La solución consiste en representar los datos en un mashup que contenga un mapa y una línea temporal. De esta forma cada incidencia queda dotada de información espacio-temporal al situar los datos sobre un mapa y asignarles un instante de tiempo. Para determinar su objetividad o subjetividad sólo habrá que crear dos apartados diferentes. En el primero se mostrará toda la información objetiva y en el segundo se mostrará el resto de la información.

Se pretende, como se ha dicho anteriormente, ordenar toda la información en el tiempo y en el espacio, para así poder disponer de una buena visualización y localización de la información. Aun así, este proyecto pretende dar un paso más.

El mashup tendrá tres perfiles para los distintos usuarios:

- Administrador: determinará los tipos de emergencia, configuración del sistema (colores, fuentes, idiomas, etc.), gestión de usuarios y gestión de incidencias.
- Usuario básico: realizará las consultas al sistema.
- Validador: su función será asegurar la veracidad de los datos introducidos y realizar la gestión de las incidencias.

Las consultas que se podrán realizar sobre el conjunto de datos serán las siguientes:

- Consulta general: se utilizará para consultar todas las incidencias ocurridas en todo el territorio. Se utiliza como paso previo a una consulta detallada y contiene a su vez dos subconsultas:
 - Línea temporal en la que aparecerán todas las incidencias indicando con ello el momento en el que ocurrieron.
 - Mapa del territorio en el que aparecerán todas las incidencias indicando con ello el lugar en el que ocurrieron.
- Consulta detallada: se utilizará para consultar una incidencia en concreto. A su vez, contendrá tres subconsultas:
 - Mapa en el que se mostrará el o las áreas en el que se produjo el incidente. Estos datos se corresponden con información objetiva.

¹⁶ Definición de Mashup: [http://es.wikipedia.org/wiki/Mashup_\(aplicación_web_híbrida\)](http://es.wikipedia.org/wiki/Mashup_(aplicación_web_híbrida))

- Mapa en el que se mostrarán todas las acciones asociadas a la incidencia. Estos datos se corresponden con información objetiva o formal.
- Mapa en el que se situarán un conjunto de imágenes, vídeos, textos, etc. asociadas a la incidencia. Estos datos se corresponderán con información subjetiva o informal.
- Comparación: se utilizará para comparar dos colecciones de datos, mostrando cada conjunto en un mapa distinto. Cada mapa tendrá asociado una línea temporal independiente que servirá para desplazarse en el tiempo:
 - En el primer mapa aparecerán todas las incidencias del primer conjunto.
 - En el segundo aparecerán todas las incidencias del segundo conjunto.

Se puede ver lo potente que puede resultar disponer de un mashup y conseguir que interprete un conjunto de datos predefinidos. Se podrán obtener de forma inmediata datos relevantes sobre la concentración, causas y otras características de los incendios forestales para poder hacer más eficiente la lucha contra este tipo de incidencias.

4.2 El proceso de desarrollo

En este capítulo se definirán todos los requisitos de capacidad y restricción. Tomando como base estos requisitos se definirán los requisitos de software, que serán la base para poder diseñar e implementar el sistema. Una vez realizada la implementación se realizarán las pruebas para garantizar que se cumplen todos los requisitos esenciales redactados.

Análisis

En la fase actual se dará forma a la aplicación web que se quiere desarrollar. Se definirán todos los requisitos que son necesarios para que la aplicación cumpla con el objetivo buscado.

Definición de requisitos

En este apartado se van a definir los requisitos basándose en las necesidades del proyecto. Los requisitos se definirán en dos tipos:

- **Requisitos de capacidad o funcionales:** especifican la funcionalidad del sistema.
- **Requisitos de restricción o no funcionales:** especifican la forma en que el sistema debe alcanzar los objetivos o realizar las distintas funciones.

Cada requisito contará con una serie de campos que lo definen. Los campos que contiene cada requisito son los siguientes:

- **Identificador:** valor único que identifica de manera unívoca al requisito.
- **Nombre:** identificador textual breve que aporta significado al requisito.
- **Prioridad:** grado de importancia del requisito en el proceso de diseño e implementación.
- **Necesidad:** grado de compromiso a la hora de incluir el requisito en el producto final.
- **Verificabilidad:** grado de dificultad a la hora de comprobar si el requisito ha sido implementado de forma correcta.
- **Descripción:** definición textual completa del requisito.

4.2...1 Requisitos de capacidad

RE-CA-01			
Nombre:	Elementos del sistema.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema almacenará incidencias. Cada incidencia podrá tener asociados acciones o reportes.		

Tabla 52. RE-CA-01 - Elementos del sistema.

RE-CA-02			
Nombre:	Consulta de incidencias.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario podrá consultar cualquier incidencia almacenada en el sistema.		

Tabla 53. RE-CA-02 – Consulta de incidencias.

RE-CA-03			
----------	--	--	--

Nombre:	Filtros en la consulta de incidencias.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de distintos filtros para poder realizar búsquedas de incidencias.		

Tabla 54. RE-CA-03 – Filtros en la consulta de incidencias.

RE-CA-04			
Nombre:	Consulta de acciones.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario podrá consultar cualquier acción almacenada en el sistema.		

Tabla 55. RE-CA-04 – Consulta de acciones.

RE-CA-05			
Nombre:	Filtros en la consulta de acciones.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de distintos filtros para poder realizar búsquedas de acciones.		

Tabla 56. RE-CA-05 – Filtros en la consulta de acciones.

RE-CA-06			
Nombre:	Consulta de reportes.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario podrá consultar cualquier reporte almacenado en el sistema.		

Tabla 57. RE-CA-06 – Consulta de reportes.

RE-CA-07			
Nombre:	Filtros en la consulta de reportes.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de distintos filtros para poder realizar búsquedas de reportes.		

Tabla 58. RE-CA-07 – Filtros en la consulta de reportes.

RE-CA-08	
Nombre:	Comparación de colección de incidencias.

Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario podrá comparar dos colecciones de incidencias.		

Tabla 59. RE-CA-08 – Comparación de colección de incidencias.

RE-CA-09			
Nombre:	Filtros en la comparación.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de distintos filtros para poder realizar comparaciones de colecciones de incidencias.		

Tabla 60. RE-CA-09 – Filtros en la comparación.

RE-CA-10			
Nombre:	Perfiles de usuario.		
Prioridad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Cada usuario del sistema deberá tener asignado un perfil, que podrá ser <i>Administrador</i> , <i>Validador</i> o <i>Usuario Básico</i> .		

Tabla 61. RE-CA-10 – Perfiles de usuario.

RE-CA-11			
Nombre:	Perfil administrador.		
Prioridad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá contar con el perfil <i>Administrador</i> . Este perfil determinará los tipos de emergencia, configuración del sistema (colores, fuentes, idiomas, etc.), gestión de usuarios y gestión de incidencias.		

Tabla 62. RE-CA-11 – Perfil administrador.

RE-CA-12			
Nombre:	Perfil validador.		
Prioridad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá contar con el perfil <i>Validador</i> . Su función será asegurar la veracidad de los datos introducidos y realizar la gestión de las incidencias.		

Tabla 63. RE-CA-12 – Perfil validador.

RE-CA-13	
Nombre:	Perfil de usuario básico.

Prioridad:	[]Alta [X]Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá contar con el perfil de Usuario Básico. Este tipo de usuario solo podrá realizar consultas al sistema.		

Tabla 64. RE-CA-13 – Perfil de usuario básico.

RE-CA-14			
Nombre:	Administración de incidencias.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir crear, eliminar y modificar cualquier incidencia, acción o reporte.		

Tabla 65. RE-CA-14 – Administración de incidencias.

RE-CA-15			
Nombre:	Administración de usuarios.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir crear, eliminar y modificar cualquier usuario.		

Tabla 66. RE-CA-15 – Administración de usuarios.

RE-CA-16			
Nombre:	Ayuda local.		
Prioridad:	[]Alta []Media [X]Baja	Necesidad:	[]Esencial []Deseable [X]Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario podrá consultar en cualquier momento una ayuda interactiva para solucionar los problemas o dudas que puedan surgir durante el uso de la aplicación.		

Tabla 67. RE-CA-16 – Ayuda local.

4.2....2 Requisitos de restricción

RE-RE-01			
Nombre:	Idioma de la interfaz.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El idioma de la interfaz de la aplicación estará disponible en inglés. La aplicación deberá poder instalar cualquier otro idioma.		

Tabla 68. RE-RE-01 – Idioma de la interfaz.

RE-RE-02			
Nombre:	Utilización de Google Maps.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación utilizará Google Maps para representar un mapa sin línea temporal.		

Tabla 69. RE-RE-02 – Utilización de Google Maps.

RE-RE-03			
Nombre:	Utilización de Timemap.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación deberá ser desarrollada mediante la tecnología Java.		

Tabla 70. RE-RE-03 – Utilización de Timemap.

RE-RE-04			
Nombre:	Lenguaje de programación Java.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación utilizará Timemap para representar un mapa con línea temporal.		

Tabla 71. RE-RE-04 – Lenguaje de programación Java.

RE-RE-05			
Nombre:	Servidor Apache Tomcat.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación deberá poder desplegarse en un servidor Apache Tomcat.		

Tabla 72. RE-RE-05 – Servidor Apache Tomcat.

RE-RE-06			
Nombre:	Base de datos MySQL.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación utilizará una base de datos MySQL para almacenar todos los datos volátiles.		

Tabla 73. RE-RE-06 – Base de datos MySQL.

RE-RE-07	
Nombre:	Navegadores válidos.

Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación deberá funcionar en el navegador Internet Explorer y Firefox.		

Tabla 74. RE-RE-07 – Navegadores válidos.

RE-RE-08			
Nombre:	Log de estado del sistema.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La aplicación creará un log de estado del sistema en el que se guardarán: <ul style="list-style-type: none"> • Trazas de errores. • Acciones realizadas. • Otros datos importantes. 		

Tabla 75. RE-RE-08 – Log de estado del sistema.

Especificación de requisitos

En este apartado se definirán los requisitos de software que surgen a raíz del análisis de los requisitos de usuario identificados en el apartado anterior.

4.2...1 Requisitos funcionales

RF-01			
Nombre:	Información asociada a una incidencia.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Cada incidencia tendrá la siguiente información: <ul style="list-style-type: none"> • Título: texto corto descriptivo de la incidencia. • Tipo: tipo de incidencia. • Subtipo: subtipo de incidencia. • Fecha de inicio: fecha en la que se inició la incidencia. • Fecha de fin: fecha en la finalizó la incidencia. • Descripción: texto largo descriptivo de la incidencia. • Acciones (información objetiva): conjunto de subincidencias asociadas a la incidencia. • Reportes (información subjetiva): conjunto de imágenes, vídeos, audios y documentos asociados a la incidencia. • Ubicación: lugar en el que se ha producido la incidencia. El lugar se mostrará en un mapa. 		

Tabla 76. RF-01 – Información asociada a una incidencia.

RF-02

Nombre:	Información asociada a una acción.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Cada acción tendrá la siguiente información: <ul style="list-style-type: none"> • Título: texto corto descriptivo de la acción. • Tipo: tipo de acción. • Subtipo: subtipo de acción. • Fecha de inicio: fecha en la que se inició la acción. • Fecha de fin: fecha en la finalizó la acción. • Descripción: texto largo descriptivo de la acción. • Ubicación: lugar en el que se ha producido la acción. El lugar se mostrará en un mapa. 		

Tabla 77. RF-02 – Información asociada a una acción.

RF-03			
Nombre:	Información asociada a un reporte.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Cada reporte tendrá la siguiente información: <ul style="list-style-type: none"> • Título: texto corto descriptivo del reporte. • Tipo: tipo de reporte. • Fecha: fecha en la que se captó el reporte. • Descripción: texto largo descriptivo del reporte. • Ubicación: lugar en el que se ha producido o captado el reporte. El lugar se mostrará en un mapa. 		

Tabla 78. RF-03 – Información asociada a un reporte.

RF-04			
Nombre:	Información asociada a un usuario.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Cada usuario tendrá la siguiente información: <ul style="list-style-type: none"> • Usuario: alias con el que se tendrá que identificar en la pantalla de inicio. • Contraseña. • Perfil: perfil de usuario. A elegir entre <i>Administrador</i>, <i>Validador</i> o <i>Usuario Básico</i>. • Nombre. • Apellidos. • Teléfono de contacto. • E-mail. 		

Tabla 79. RF-04 – Información asociada a un usuario.

RF-05			
Nombre:	Consulta de incidencias mediante un mapa.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un mapa en el que aparecerán todas las incidencias de acuerdo con su ubicación. Además, el sistema dispondrá de una línea temporal en el que aparecerán todas las incidencias ordenadas cronológicamente.		

Tabla 80. RF-05 – Consulta de incidencias mediante un mapa.

RF-06			
Nombre:	Consulta de incidencias mediante una lista.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de una lista en el que aparecerán todas las incidencias.		

Tabla 81. RF-06 – Consulta de incidencias mediante una lista.

RF-07			
Nombre:	Filtro de texto exclusivo en la consulta de incidencias.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de texto en el que el usuario indicará que palabras deben estar contenidas en el título de la incidencia. Si la o las palabras escritas no existen en el título de la incidencia, ésta no aparecerá ni el mapa ni en la línea temporal.		

Tabla 82. RF-07 – Filtro de texto exclusivo en la consulta de incidencias.

RF-08			
Nombre:	Filtro de texto inclusivo en la consulta de incidencias.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de texto en el que el usuario indicará que palabras deben estar contenidas en el título de la incidencia. Si la o las palabras escritas existen en el título de la incidencia, ésta aparecerá resaltada con un recuadro.		

Tabla 83. RF-08 – Filtro de texto inclusivo en la consulta de incidencias.

RF-09	
Nombre:	Filtro de tipos en la consulta de incidencias.

Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de tipos en el que el usuario seleccionará que tipos y/o subtipos se mostrarán.		

Tabla 84. RF-09 – Filtro de tipos en la consulta de incidencias.

RF-10			
Nombre:	Filtro por fecha en la consulta de incidencias.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro por fecha en el que el usuario seleccionará la fecha en la que se centrará la línea temporal.		

Tabla 85. RF-10 – Filtro por fecha en la consulta de incidencias.

RF-11			
Nombre:	Consulta de acciones.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un mapa por cada incidencia en el que aparecerán todas las acciones de acuerdo con su ubicación. Además, el sistema dispondrá de una lista en la que aparecerán todas las acciones.		

Tabla 86. RF-11 – Consulta de acciones.

RF-12			
Nombre:	Filtro de texto exclusivo en la consulta de acciones.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de texto en el que el usuario indicará que palabras deben estar contenidas en el título de la acción. Si la o las palabras escritas no existen en el título de la acción, ésta no aparecerá ni el mapa ni en la línea temporal.		

Tabla 87. RF-12 – Filtro de texto exclusivo en la consulta de acciones.

RF-13			
Nombre:	Filtro de texto inclusivo en la consulta de acciones.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		

Descripción	El sistema deberá disponer de un filtro de texto en el que el usuario indicará que palabras deben estar contenidas en el título de la acción. Si la o las palabras escritas existen en el título de la acción, ésta aparecerá resaltada con un recuadro.
--------------------	--

Tabla 88. RF-13 – Filtro de texto inclusivo en la consulta de acciones.

RF-14			
Nombre:	Filtro por fecha en la consulta de acciones.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro por fecha en el que el usuario seleccionará la fecha en la que se centrará la línea temporal.		

Tabla 89. RF-14 – Filtro por fecha en la consulta de acciones.

RF-15			
Nombre:	Consulta de reportes.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un mapa por cada incidencia en el que aparecerán todos los reportes de acuerdo con su ubicación. Además, el sistema dispondrá de un visor en la que aparecerán todos los reportes, mostrando una pequeña imagen de acuerdo a su tipo.		

Tabla 90. RF-15 – Consulta de reportes.

RF-16			
Nombre:	Filtro de texto exclusivo en la consulta de reportes.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de texto en el que el usuario indicará que palabras deben estar contenidas en el título del reporte. Si la o las palabras escritas no existen en el título del reporte, ésta no aparecerá ni el mapa ni en la línea temporal.		

Tabla 91. RF-16 – Filtro de texto exclusivo en la consulta de reportes.

RF-17			
Nombre:	Filtro de texto inclusivo en la consulta de reportes.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		

Descripción	El sistema deberá disponer de un filtro de texto en el que el usuario indicará que palabras deben estar contenidas en el título del reporte. Si la o las palabras escritas existen en el título del reporte, ésta aparecerá resaltada con un recuadro.
--------------------	--

Tabla 92. RF-17 – Filtro de texto inclusivo en la consulta de acciones.

RF-18			
Nombre:	Filtro por fecha en la consulta de reportes.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro por fecha en el que el usuario seleccionará la fecha en la que se centrará la línea temporal.		

Tabla 93. RF-18 – Filtro por fecha en la consulta de acciones.

RF-19			
Nombre:	Comparación de colección de incidencias.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de dos mapas en los que se mostrarán dos colecciones de datos. Cada mapa tendrá una línea temporal independiente.		

Tabla 94. RF-19 – Comparación de colección de incidencias.

RF-20			
Nombre:	Filtro de texto exclusivo en la comparación.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de texto por cada mapa en el que el usuario indicará que palabras deben estar contenidas en el título de la incidencia. Si la o las palabras escritas no existen en el título de la incidencia, ésta no aparecerá ni el mapa ni en la línea temporal.		

Tabla 95. RF-20 – Filtro de texto exclusivo en la comparación.

RF-21			
Nombre:	Filtro de texto inclusivo en la comparación.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de texto por cada mapa en el que el usuario indicará que palabras deben estar contenidas en el título de la incidencia. Si la o las palabras escritas existen en el título de la incidencia, ésta aparecerá resaltada con un recuadro.		

Tabla 96. RF-21 – Filtro de texto inclusivo en la comparación.

RF-22			
Nombre:	Filtro de tipos en la comparación.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro de tipos por cada mapa en el que el usuario seleccionará que tipos y/o subtipos se mostrarán.		

Tabla 97. RF-22 – Filtro de tipos en la comparación.

RF-23			
Nombre:	Filtro por fecha en la comparación.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá disponer de un filtro por fecha por cada mapa en el que el usuario seleccionará la fecha en la que se centrará la línea temporal.		

Tabla 98. RF-23 – Filtro por fecha en la comparación.

RF-24			
Nombre:	Perfiles de usuario.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Cada usuario del sistema deberá tener asignado un perfil, que podrá ser <i>Administrador</i> , <i>Validador</i> o <i>Usuario Básico</i> .		

Tabla 99. RF-24 – Perfiles de usuario.

RF-25			
Nombre:	Perfil administrador.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá contar con el perfil <i>Administrador</i> . Este perfil determinará los tipos de emergencia, configuración del sistema (colores, fuentes, idiomas, etc.), gestión de usuarios y gestión de incidencias.		

Tabla 100. RF-25 – Perfil administrador.

RF-26			
Nombre:	Perfil validador.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá contar con el perfil <i>Validador</i> . Su función será asegurar la veracidad de los datos introducidos y realizar la gestión de las incidencias.		

Tabla 101. RF-26 – Perfil validador.

RF-27			
Nombre:	Perfil de usuario básico.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá contar con el perfil de Usuario Básico. Este tipo de usuario solo podrá realizar consultas al sistema.		

Tabla 102. RF-27 – Perfil de usuario básico.

RF-28			
Nombre:	Guardar incidencia.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir guardar una nueva incidencia en el sistema.		

Tabla 103. RF-28 – Guardar incidencia.

RF-29			
Nombre:	Guardar acción.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir guardar una nueva acción, asociada a una incidencia, en el sistema.		

Tabla 104. RF-29 – Guardar acción.

RF-30			
Nombre:	Guardar reporte.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir guardar un nuevo reporte, asociado a una incidencia, en el sistema.		

Tabla 105. RF-30 – Guardar reporte.

RF-31			
Nombre:	Guardar ubicación.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir guardar la ubicación de cualquier elemento indicando el punto o área mediante un mapa.		

Tabla 106. RF-31 – Guardar ubicación.

RF-32			
Nombre:	Administración de incidencias.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir eliminar y modificar cualquier incidencia, acción y reporte guardado.		

Tabla 107. RF-32 – Administración de incidencias.

RF-33			
Nombre:	Administración de usuarios.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá permitir eliminar y modificar cualquier usuario registrado en el sistema.		

Tabla 108. RF-33 – Administración de usuarios.

RF-34			
Nombre:	Ayuda local.		
Prioridad:	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja	Necesidad:	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input checked="" type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario podrá consultar en cualquier momento una ayuda interactiva para solucionar los problemas o dudas que puedan surgir durante el uso de la aplicación.		

Tabla 109. RF-34 – Ayuda local.

4.2...2 Requisitos de datos

Estos requisitos especifican los valores de cada elemento del sistema.

RD-01			
Nombre:	Datos de una incidencia.		
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Verificabilidad	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja		
Estabilidad:	Media.		

Descripción	<p>Cada incidencia tendrá la siguiente información:</p> <ul style="list-style-type: none"> • Título: texto, 60 caracteres. • Tipo: texto, 30 caracteres. • Subtipo: texto, 30 caracteres. • Fecha de inicio: dd-mm-aaaa, 10 caracteres. • Fecha de fin: dd-mm-aaaa, 10 caracteres. • Descripción: texto, 200 caracteres. • Longitud: 30 caracteres. • Latitud: 30 caracteres.
--------------------	---

Tabla 110. RD-01 – Datos de una incidencia.

RD-02			
Nombre:	Datos de una acción.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Media.		
Descripción	<p>Cada acción tendrá la siguiente información:</p> <ul style="list-style-type: none"> • Título: texto, 60 caracteres. • Tipo: texto, 40 caracteres. • Subtipo: texto, 40 caracteres. • Fecha de inicio: dd-mm-aaaa, 10 caracteres. • Fecha de fin: dd-mm-aaaa, 10 caracteres. • Descripción: texto, 200 caracteres. • Longitud: 30 caracteres. • Latitud: 30 caracteres. 		

Tabla 111. RD-02 – Datos de una acción.

RD-03			
Nombre:	Datos de un reporte.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Media.		
Descripción	<p>Cada reporte tendrá la siguiente información:</p> <ul style="list-style-type: none"> • Título: texto, 60 caracteres. • Tipo: texto, 12 caracteres. • Fecha: dd-mm-aaaa, 10 caracteres. • Descripción: texto, 200 caracteres. • Longitud: 30 caracteres. • Latitud: 30 caracteres. 		

Tabla 112. RD-03 – Datos de un reporte.

RD-04			
Nombre:	Datos de un usuario.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		

Estabilidad:	Media.
Descripción	<p>Cada usuario tendrá la siguiente información:</p> <ul style="list-style-type: none">• Usuario: texto, 20 caracteres.• Contraseña: texto, 20 caracteres.• Perfil: texto, 30 caracteres.• Nombre: texto, 20 caracteres.• Apellidos: texto, 60 caracteres.• Teléfono de contacto: numérico, 15 números.• E-mail: texto, 30 caracteres.

Tabla 113. RD-04 – Datos de un usuario.

4.2...3 Requisito de interfaz

Estos requisitos comprenden las necesidades de comunicación, tanto en redes, hardware y software.

RI-01			
Nombre:	Comunicación de la aplicación con la base de datos.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema almacenará toda la información que necesite en una base de datos a través de un conector. Deberá tener definidas todas las operaciones para acceder y modificar los distintos valores.		

Tabla 114. RI-01 – Comunicación de la aplicación con la base de datos.

4.2...4 Requisitos operacionales

Estos requisitos especifican cómo debe ejecutarse el sistema y cómo se comunicará con los operadores humanos.

RO-01			
Nombre:	Comunicación del usuario con el sistema.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Se accederá al sistema a través de un navegador web compatible. El usuario podrá realizar la administración y consultas de incidencias, reportes y acciones.		

Tabla 115. RO-01 – Comunicación del usuario con el sistema.

RO-02			
Nombre:	Ejecución del sistema.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema se ejecutará, íntegramente, en un mismo servidor.		

Tabla 116. RO-02 – Ejecución del sistema.

4.2...5 Requisitos de verificación

Conjunto de requisitos que especifican las restricciones de cómo será verificado el software.

RV-01			
Nombre:	Simulación de la aplicación.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[X]Alta []Media []Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		

Descripción	Se comprobará el correcto funcionamiento de la aplicación mediante una simulación en un entorno lo más real posible. Esta simulación se llevará a cabo con un conjunto de datos válidos para incidencias, acciones, reportes y usuarios.
--------------------	--

Tabla 117. RV-01 – Simulación de la aplicación.

4.2...6 Requisitos de pruebas de aceptación

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

RPA-01			
Nombre:	Funcionamiento completo.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta [X]Media []Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá funcionar de forma correcta, ofreciendo todas las funciones definidas en los requisitos de software.		

Tabla 118. RPA-01 – Funcionamiento completo.

4.2...7 Requisitos de documentación

Estos requisitos especifican la documentación propia del proyecto.

RD-01			
Nombre:	Manual de usuario.		
Prioridad:	[]Alta [X]Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá funcionar de forma correcta, ofreciendo todas las funciones definidas en los requisitos de software.		

Tabla 119. RD-01 – Manual de usuario.

4.2...8 Requisitos de seguridad

Se refiere a los requisitos que eviten amenazas externas o por parte del usuario.

RSG-01			
Nombre:	Validación de campos.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá comprobar que los datos recibidos son del tipo y de la longitud definida.		

Tabla 120. RSG-01 – Validación de campos.

RSG-02			
Nombre:	Autenticación de usuarios.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El usuario deberá identificarse para poder acceder al sistema mediante su nombre de usuario y contraseña.		

Tabla 121. RSG-02 – Autenticación de usuarios.

4.2...9 Requisitos de portabilidad

Conjunto de requisitos que especifican la facilidad de modificar el software para ser ejecutado en otras máquinas o sistemas operativos.

RP-01			
Nombre:	Lenguaje de programación.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema deberá estar implementado mediante la tecnología Java.		

Tabla 122. RP-01 – Lenguaje de programación.

RP-02			
Nombre:	Base de datos.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Todas las incidencias, acciones, reportes y usuarios deberán ser guardados en una base de datos.		

Tabla 123. RP-02 – Base de datos.

RP-03			
Nombre:	MySQL.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema utilizará una base de datos MySQL.		

Tabla 124. RP-03 – MySQL.

RP-04			
Nombre:	Máquina virtual Java.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		

Descripción	El servidor dónde esté alojado la aplicación deberá tener instalada la máquina virtual de Java.
--------------------	---

Tabla 125. RP-04 – Máquina virtual Java.

RP-05			
Nombre:	Datos TimeMaps.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	Los datos de los TimeMaps deberán ser guardados en documentos XML.		

Tabla 126. RP-05 – Datos TimeMaps.

RP-06			
Nombre:	Codificación UTF8.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	La codificación de los caracteres de la base de datos será UTF-8.		

Tabla 127. RP-06 – Codificación UTF-8.

4.2...10 Requisitos de calidad

Estos requisitos indican que puntos de fiabilidad serán esperados por parte del sistema.

RCAL-01			
Nombre:	Concurrencia.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema será concurrente. Deberá ser capaz de atender a las peticiones de más de un usuario.		

Tabla 128. RCAL-01 – Concurrencia.

4.2...11 Requisitos de mantenimiento

Estos requisitos especifican cómo de fácil es reparar los fallos y adaptar el software a los nuevos requerimientos.

RM-01			
Nombre:	Log de estado del sistema.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		

Descripción	El sistema generará archivos con información sobre el estado del sistema. Estos archivos ayudarán a realizar un mantenimiento y corrección de errores más preciso.
--------------------	--

Tabla 129. RM-01 – Log de estado del sistema.

RM-02			
Nombre:	Log errores internos.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[X]Esencial []Deseable []Opcional
Verificabilidad	[]Alta []Media [X]Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema generará archivos donde guardará toda la información cuando se produce un error. Deberá almacenar la fecha, hora y traza del error.		

Tabla 130. RM-02 – Log errores internos.

RM-03			
Nombre:	Utilización de patrones en la implementación.		
Prioridad:	[X]Alta []Media []Baja	Necesidad:	[]Esencial [X]Deseable []Opcional
Verificabilidad	[X]Alta []Media []Baja		
Estabilidad:	Alta. Durante toda la vida del sistema.		
Descripción	El sistema será diseñado recurriendo al uso de patrones de diseño. De esta manera el código será más flexible a cambios y actualizaciones.		

Tabla 131. RM-03 – Utilización de patrones en la implementación.

Diseño

En este apartado se realizará una descomposición del sistema en componentes más pequeños, denominados subsistemas.

Cuanto mayor sea la descomposición del sistema, más sencillos serán los subsistemas resultantes, aunque de esta forma aumenta la complejidad de las interfaces que los interrelacionan. Por ello, se optará por una solución de compromiso.

En este proyecto se empleará como criterio de división la funcionalidad. De esta manera, cada subsistema cubrirá unas necesidades de la aplicación.

Método de diseño

Para realizar el diseño del sistema se utilizará el método de diseño UML y como herramienta de diseño se utilizará la herramienta Altova UModel 2011.

Se realizará una descomposición en componentes orientada a objetos mostrando la estructura principal del sistema.

Diseño de sistema

Para independizar el modelo de datos de las operaciones, y todo esto a su vez de la presentación al usuario, se va a implementar el patrón arquitectónico modelo-vista-controlador (MVC), con un añadido para implementar la comunicación con la infraestructura externa, es decir, la BBDD que es de donde la aplicación se encargará de obtener la información necesaria.

Este modelo divide el sistema en cuatro subsistemas que son los siguientes:

- Vista: esta capa se encargará de mostrar la interfaz gráfica de la aplicación al usuario. Ofreciéndole la posibilidad de realizar todas las operaciones especificadas en los requisitos de software.
- Controlador: capa encargada de reaccionar ante los eventos ocurridos en la aplicación, ya sean acciones del usuario, cambios en el negocio o en la vista.
- Modelo: esta capa se encargará de la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- Infraestructura: capa a través de la cual la aplicación se va a comunicar con la infraestructura externa, en este caso con la base de datos, donde se almacena la información sobre los parámetros de configuración.

Los subsistemas se relacionan como se muestra a continuación:

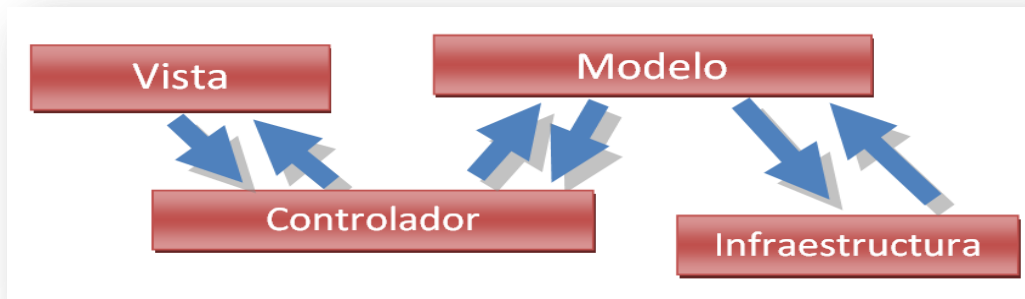


Figura 13. MVC + I.

A continuación se especificarán los subsistemas de los que se compone la aplicación:

4.2...1 Subsistema Vista

SUB-01	Vista
Funciones:	<ul style="list-style-type: none"> • Provee a la aplicación de una interfaz gráfica. • Permite al usuario comunicarse con el sistema. • Varía en función de las peticiones del usuario.

Tabla 132. Subsistema Vista.

Los módulos del subsistema Vista son los siguientes:

- **Módulo de Consultas:** este módulo contendrá aquella funcionalidad necesaria para realizar consultas al sistema.
- **Módulo de Administración:** este módulo contendrá aquella funcionalidad necesaria para realizar la administración de incidencias y usuarios.

4.2...2 Subsistema Controlador

SUB-02	Controlador
Funciones:	<ul style="list-style-type: none"> • Procesa las peticiones recibidas por el subsistema Vista. • Modifica el modelo de acuerdo con la operación en curso.

Tabla 133. Subsistema Controlador.

Los módulos del subsistema controlador son los siguientes:

- **Módulo JSF:** este módulo contendrá aquella funcionalidad necesaria para comunicar al subsistema vista con el sistema. Reaccionará ante la consulta, editado y borrado de incidencias y usuarios.

4.2...3 Subsistema Negocio

SUB-03	Modelo
Funciones:	<ul style="list-style-type: none"> Se encarga de proporcionar la información al sistema. Se comunica con el subsistema infraestructura para guardar, eliminar o modificar datos en la base de datos.

Tabla 134. Subsistema Modelo.

Los módulos del subsistema modelo son los siguientes:

- **Módulo KML:** desde este módulo se gestionará toda la información relacionada con los documentos KML necesarios para alimentar a los TimeMap.
- **Módulo Gestión:** desde este módulo se gestionarán las incidencias y usuarios.
- **Módulo DAO:** módulo que independizará los datos de la BD que los almacena.

4.2...4 Subsistema Infraestructura

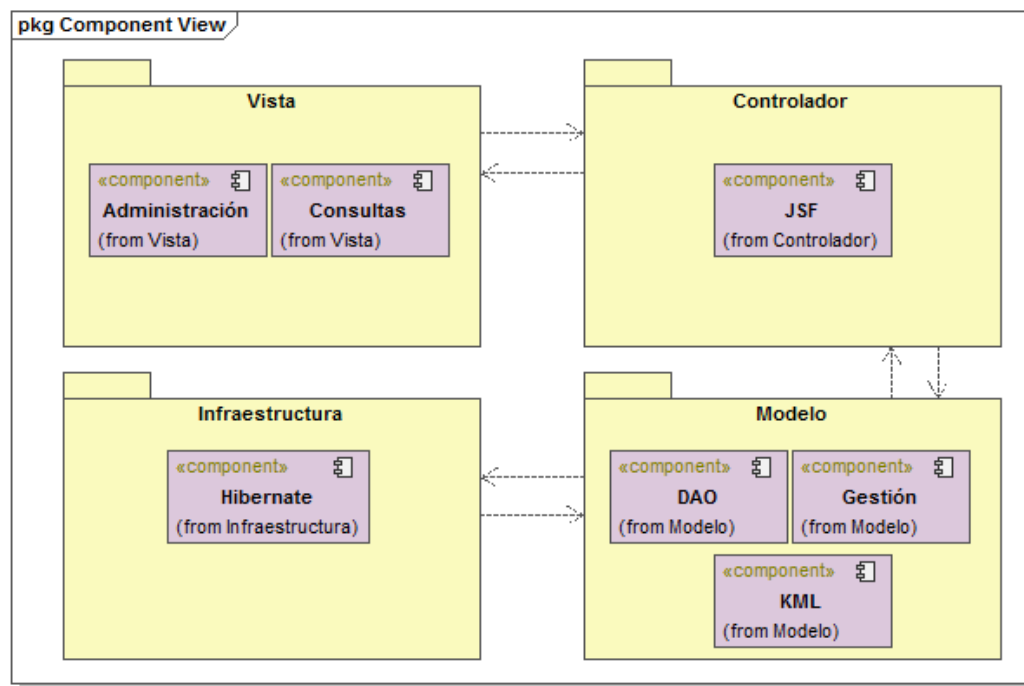
SUB-04	Infraestructura
Funciones:	<ul style="list-style-type: none"> Se encarga de comunicarse con la base de datos para obtener, modificar y borrar información.

Tabla 135. Subsistema Infraestructura.

El módulo del subsistema infraestructura son los siguientes:

- **Módulo Hibernate:** desde este módulo se gestionará toda la comunicación con la BBDD.

A continuación se muestra el diagrama de los subsistemas que se ha explicado:



Generated by UModel

www.altova.com

Figura 14. Diagrama de subsistemas.

Diseño detallado

En este apartado se realizará un diseño detallo del sistema tomando como base el diagrama de subsistemas mostrado en la Figura 14. Diagrama de subsistemas.. Se describirá cada una de las clases que han surgido, identificando las responsabilidades que tienen asociadas, sus atributos y las relaciones entre ellas. Para disponer de una visión general del diagrama de clases se presenta la siguiente figura:

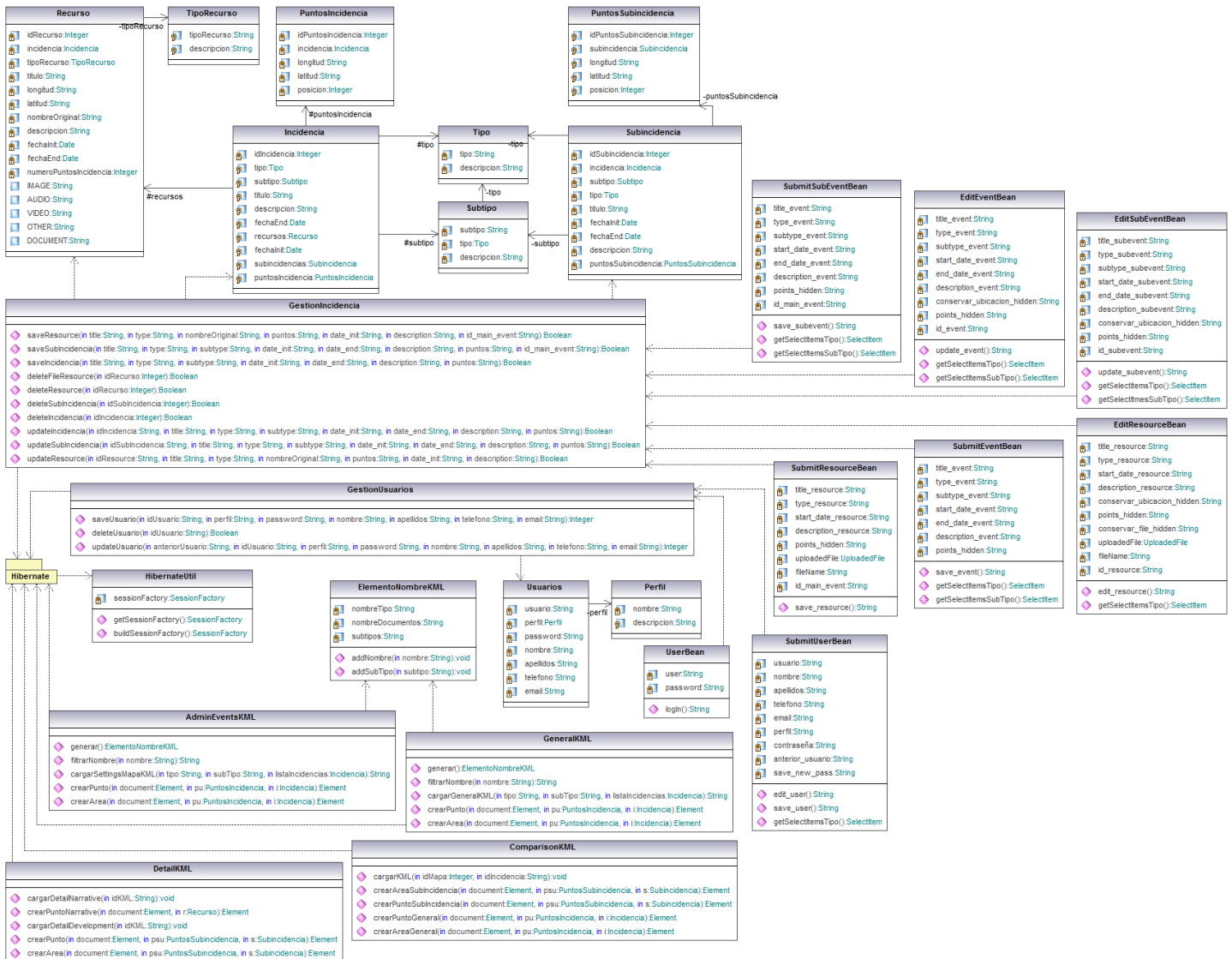


Figura 15. Diagrama de clases general.

A continuación se describe la capa Vista, Controlador, Modelo e Infraestructura con más detalle.

4.2...1 Capa vista

Dado que se trata de una aplicación web, la capa Vista estará formada por el conjunto de todas las páginas JSP con las que el usuario del sistema realizará todas las operaciones definidas.

Hay que señalar que se utilizará la tecnología JSF para realizar el envío y recepción de datos con el sistema, por lo que la capa *Vista* enviará los datos a la capa *Controlador* haciendo uso del servlet FacesServlet de JSF.

4.2...2 Capa controlador

La capa controlador del sistema está muy bien definida ya que se utiliza la tecnología JSF. Para que JSF pueda funcionar, se necesita la existencia de un servlet llamado FacesServlet. Este servlet se encargará de interceptar todos los flujos de comunicación que existirán entre los elementos JSF y sus respectivos Beans, es decir, se encargará de la comunicación entre la capa *Vista* y la capa *Modelo*.

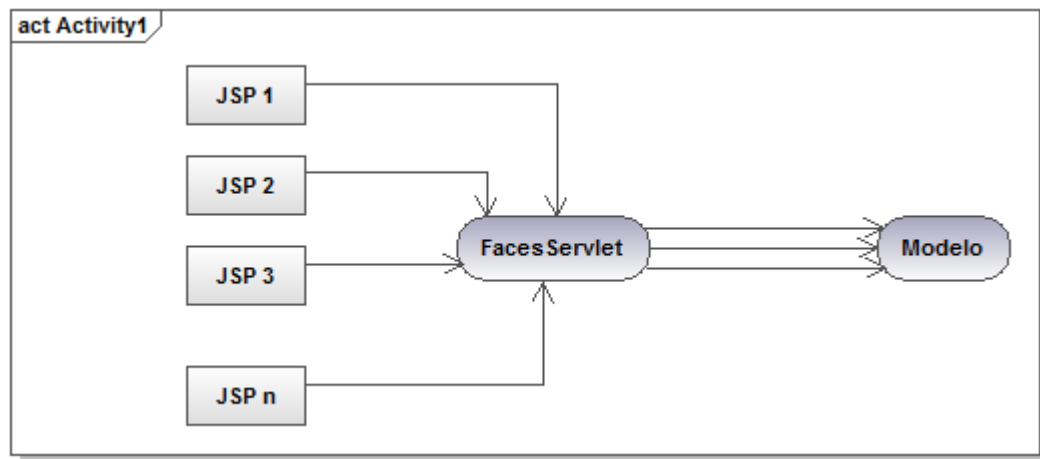


Figura 16. Comunicación Vista-Controlador-Modelo.

En la anterior figura (Figura 16) se puede ver que todas las peticiones de todos los JSP (capa *Vista*) son dirigidas a FacesServlet (capa *Controlador*), y éste las redirige a cada elemento dentro de la capa *Modelo*. De esta manera todo el flujo de información pasa por la capa *Controlador*.

4.2...3 Capa modelo

La capa modelo contiene todos aquellos elementos relacionados con el modelo de negocio de la aplicación. Atendiendo al modelo de subsistemas (Figura 14), la capa modelo contiene los siguientes elementos:

- Gestión: este elemento a su vez contiene la gestión de usuarios y la gestión de incidencias.
- KML: se encarga de la gestión de los documentos KML que utilizarán los elementos que implementan Timemap.
- DAO: contiene todas las clases necesarias que sirven para acceder a los datos guardados en la base de datos.

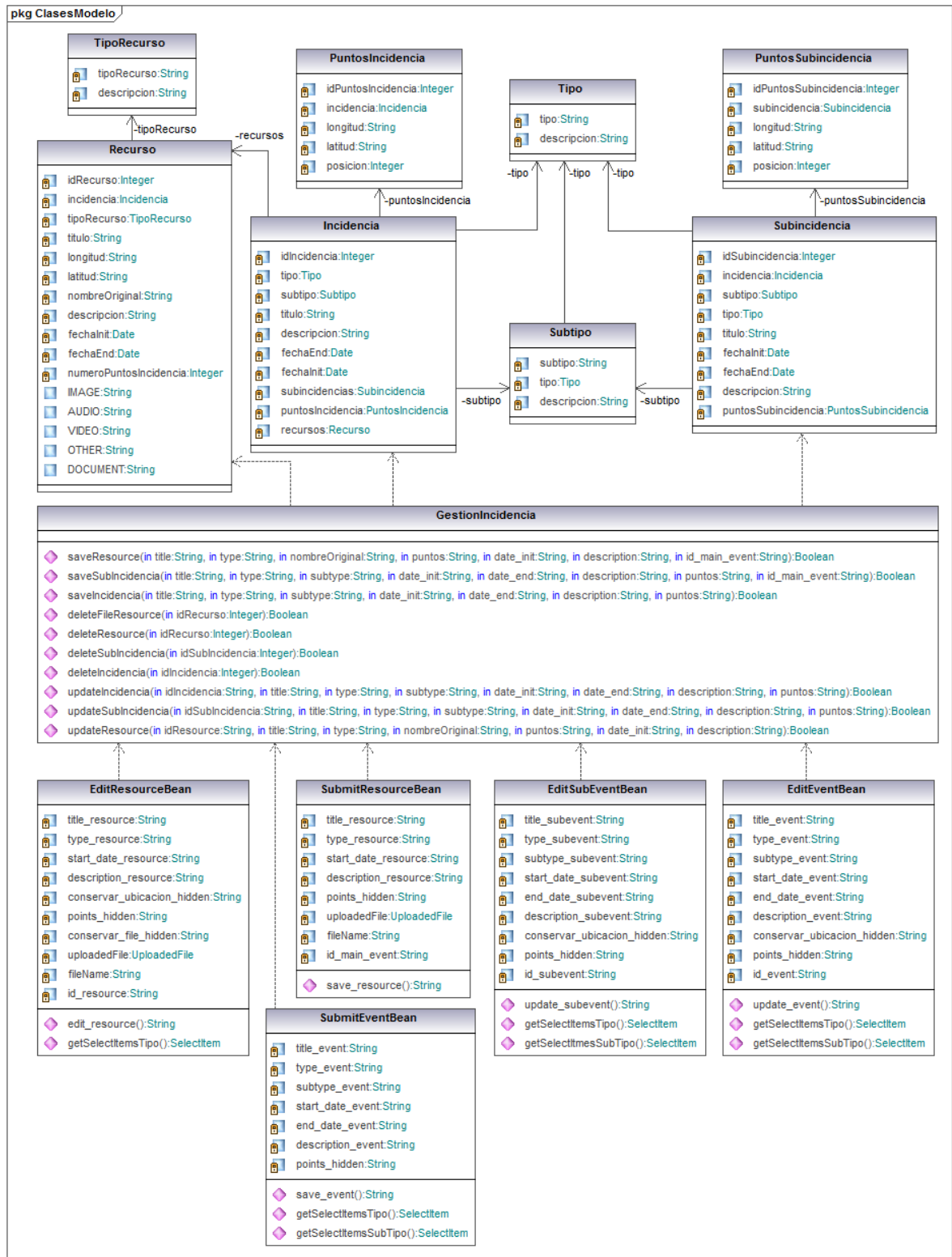


Figura 17. Modelo de clases de la gestión de incidencias.

En la Figura 17. Modelo de clases de la gestión de incidencias. se muestra todo el conjunto de clases que pertenecen a la capa *Modelo* relacionadas con la gestión de las incidencias. Las clases que pertenecen al subsistema DAO son:

- TipoRecurso: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de tipos de recurso.
- Recurso: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de recursos.
- Incidencia: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de incidencias.
- PuntosIncidencia: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de puntos de incidencia.
- Tipo: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de tipos de incidencia.
- Subtipo: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de subtipos de incidencia.
- Subincidencia: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de subincidencias, corresponde con las acciones de una incidencia
- PuntosSubincidencia: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de puntos de subincidencia, corresponde con los puntos de la ubicación de las acciones de una incidencia.

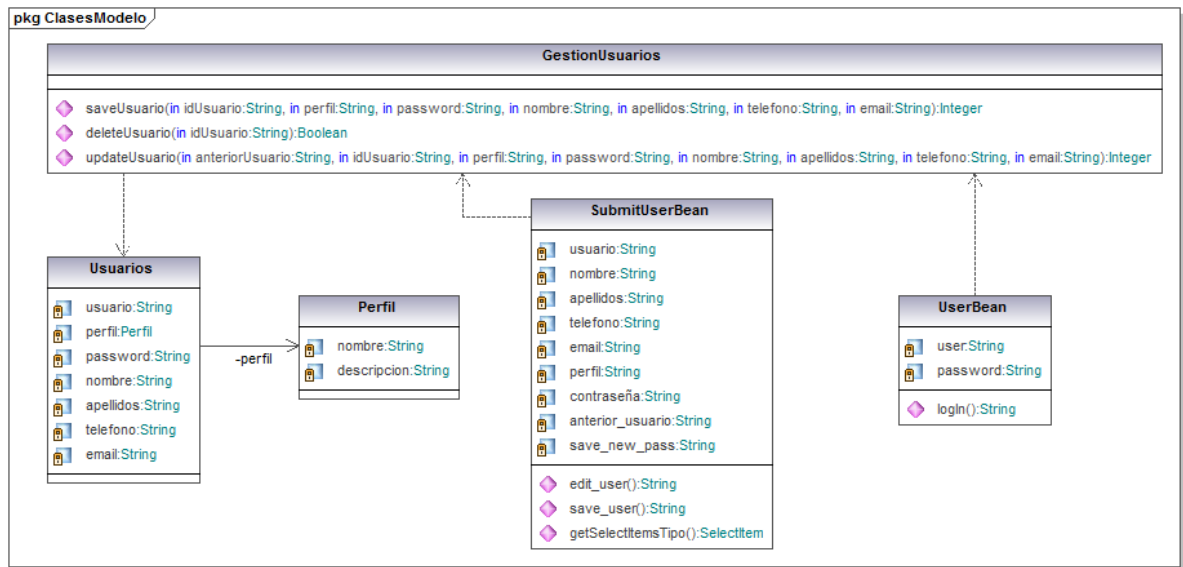
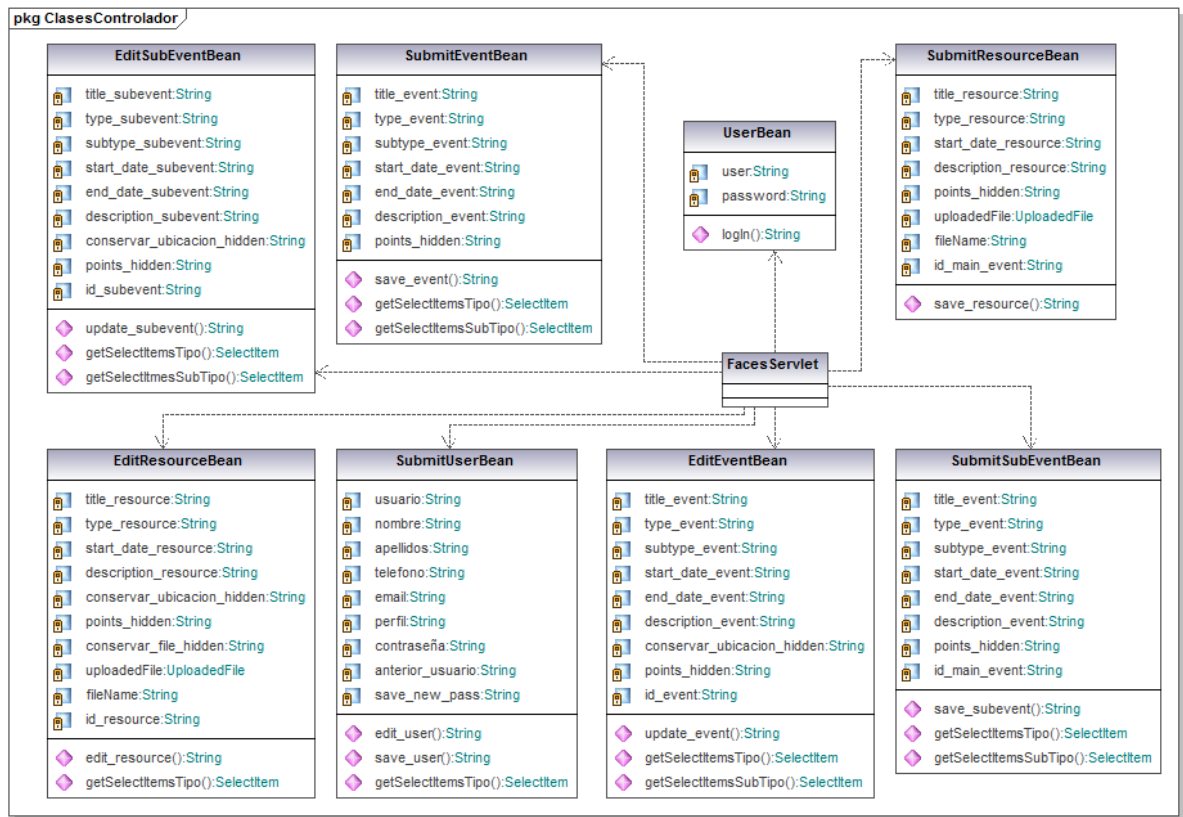


Figura 18. Modelo de clases relacionado con la gestión de usuarios.

En la Figura 18. Modelo de clases relacionado con la gestión de usuarios. se muestra el conjunto de clases de la capa *Modelo* que se encarga de la gestión de los usuarios. Las clases que pertenecen al subsistema DAO son:

- Usuarios: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de usuarios.
- Perfil: clase que sirve para recuperar, actualizar o crear una entrada de la tabla de tipos de usuario.



Generated by UModel

www.altova.com

Figura 19. Relación capa Controlador con capa Modelo.

En la Figura 19. Relación capa Controlador con capa Modelo. se muestra la comunicación entre la capa *Controlador* y la capa *Modelo*. La clase *FacesServlet* será el objeto que encaminará todas las peticiones realizadas por la capa *Vista*, que estará formada por el conjunto de todas las páginas JSP. El servlet *FacesServlet* se encarga de recoger los datos enviados por las páginas JSP e introducirlos en los campos de cada Bean.

Para poder mostrar las clases del subsistema KML es necesario mostrar su relación con la capa *Infraestructura*.

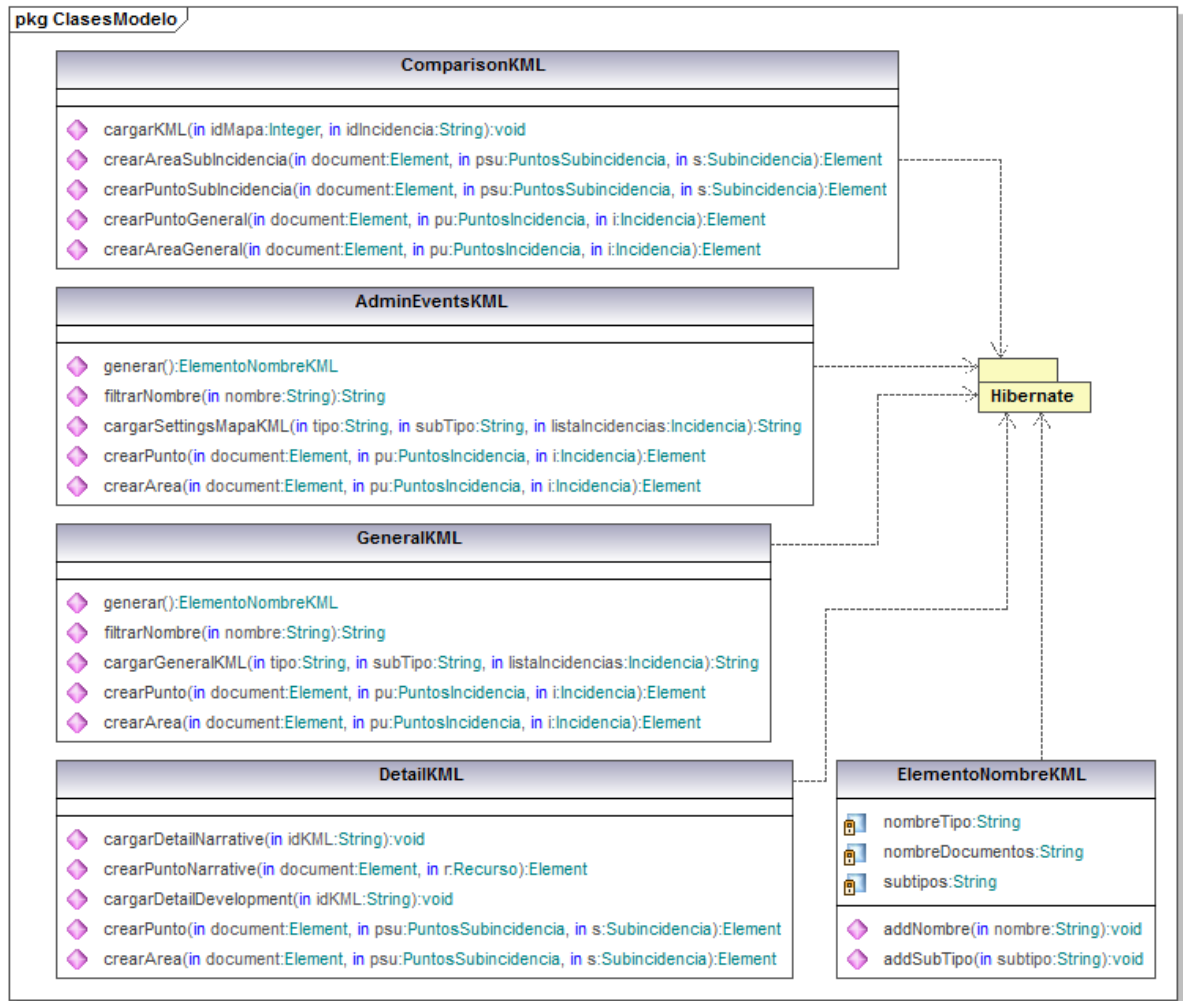


Figura 20. Modelo de clases de la gestión KML con la capa Infraestructura.

En la Figura 20. Modelo de clases de la gestión KML con la capa Infraestructura. se muestra el conjunto de clases encargadas con la gestión de los documentos KML y la capa *Infraestructura*, que se muestra como el paquete *Hibernate*. La función de cada clase es la siguiente:

- **ComparisonKML**: se encarga de generar los archivos KML necesarios de donde tomarán los datos los Timemap de la comparación de incidencias.
- **AdminEventsKML**: se encarga de generar los archivos KML necesarios de donde tomarán los datos los Timemap de la administración de incidencias.
- **GeneralKML**: se encarga de generar los archivos KML necesarios de donde tomarán los datos los Timemap de la pantalla general.
- **DetailKML**: se encarga de generar los archivos KML necesarios de donde tomarán los datos los Timemap que muestran en detalle cada incidencia.
- **ElementoNombreKML**: almacena los datos necesarios para que los Timemap puedan localizar los documentos KML.

A continuación se explican con detalle todas las clases de la capa Modelo.

4.2...3.1 Clases de gestión

	GestionIncidencia
Responsabilidades	Contiene todas las operaciones necesarias para gestionar una incidencia.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean saveResource(String title, String type, String nombreOriginal, String puntos, String date_init, String description, String id_main_event): método que se encarga de guardar un recurso enviado por el usuario. • public boolean saveSubIncidencia(String title, String type, String subtype, String date_init, String date_end, String description, String puntos, String id_main_event): método que se encarga de guardar una acción enviada por el usuario. • public boolean saveIncidencia(String title, String type, String subtype, String date_init, String date_end, String description, String puntos): método que se encarga de guardar una incidencia enviada por el usuario. • public boolean deleteFileResource(int idRecurso): método que se encarga de borrar el documento de un recurso. • public boolean deleteResource(int idRecurso): método que se encarga de borrar un recurso. • public boolean deleteSubIncidencia(int idSubIncidencia): método que se encarga de borrar una acción. • public boolean deleteIncidencia(int idIncidencia): método que se encarga de borrar una incidencia. • public boolean updateIncidencia(String idIncidencia, String title, String type, String subtype, String date_init, String date_end, String description, String puntos): método que se encarga de actualizar los datos de una incidencia. • public boolean updateSubIncidencia(String idSubIncidencia, String title, String type, String subtype, String date_init, String date_end, String description, String puntos): método que se encarga de actualizar los datos de una acción. • public boolean updateResource(String idResource, String title, String type, String nombreOriginal, String puntos, String date_init, String description): método que se encarga de actualizar los datos de un recurso.

Tabla 136. Clase GestionIncidencia.

	GestionUsuarios
Responsabilidades	Contiene todas las operaciones necesarias para gestionar un usuario.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean saveUsuario(String idUsuario, String perfil, String password, String nombre, String apellidos, String telefono, String email): método que se encarga de guardar un nuevo usuario. • public boolean deleteUsuario(String idUsuario): método que se encarga de borrar un usuario. • public boolean updateUsuario(String usuarioAnterior, String idUsuario, String perfil, String password, String nombre, String apellidos, String telefono, String email): método que se encarga de actualizar los datos de

un usuario.

Tabla 137. Clase GestionUsuarios.

	GeneralKML
Responsabilidades	Contiene todas las operaciones necesarias para gestionar los documentos KML que necesita la vista general.
Atributos	<ul style="list-style-type: none"> • private List<ElementoNombreKML> listaKML: contiene la lista de los documentos KML que necesitarán los Timemap de la vista general.
Operaciones	<ul style="list-style-type: none"> • public List<ElementoNombreKML> generar(): genera los documentos KML, devolviendo la lista de documentos KML creados. • public String filtrarNombre(String nombre): filtra los nombres de los documentos para eliminar caracteres especiales. • private String cargarGeneralKML(String tipo, String subtipo, List<Incidencia> listaIncidencias): genera el documento KML para el tipo y subtipo pasados por argumento. • private Element crearPunto(Element document, List<PuntosIncidencia> pu, Incidencia i): crea un elemento Element con la información de un punto. • private Element crearArea(Element document, List<PuntosIncidencia> pu, Incidencia i): crea un elemento Element con la información de un área.

Tabla 138. Clase GeneralKML.

	DetailKML
Responsabilidades	Contiene todas las operaciones necesarias para gestionar los documentos KML que necesita la vista general.
Atributos	No tiene atributos.
Operaciones	<ul style="list-style-type: none"> • private void cargarDetailNarrative(String idKML): genera los documentos KML para los timemap de los reportes de la vista detalle. • private Element crearPuntoNarrative(Element document, Recurso r): crea un elemento Element con la información de un punto. • private void cargarDetailDevelopment(String idKML): genera los documentos KML para los timemap de las acciones de la vista detalle. • private Element crearPunto(Element document, List<PuntosSubincidencia> psu, Subincidencia s): crea un elemento Element con la información de un punto. • private Element crearArea(Element document, List<PuntosSubincidencia> psu, Subincidencia s): crea un elemento Element con la información de un área.

Tabla 139. Clase DetailKML.

	ComparisonKML
Responsabilidades	Contiene todas las operaciones necesarias para gestionar los documentos KML que necesita la vista de comparación.
Atributos	No tiene atributos.
Operaciones	<ul style="list-style-type: none"> • public void cargarKML(int idMapa, String idIncidencia): genera los documentos KML para los timemap de la vista de comparación. • private Element crearAreaSubIncidencia(Element document, List<PuntosSubincidencia> psu, Subincidencia s): crea un elemento Element con la información de un área. • private Element crearPuntoSubIncidencia(Element document, List<PuntosSubincidencia> psu, Subincidencia s): crea un elemento

	<p>Element con la información de un punto.</p> <ul style="list-style-type: none"> • private Element crearPuntoGeneral(Element document, List<PuntosIncidencia> pu, Incidencia i): crea un elemento Element con la información de un punto.
--	--

Tabla 140. Clase ComparisonKML.

	AdminEventsKML
Responsabilidades	Contiene todas las operaciones necesarias para gestionar los documentos KML que necesita la administración de incidencias, acciones y reportes.
Atributos	<ul style="list-style-type: none"> • private List<ElementoNombreKML> listaKML: contiene la lista de nombres de documentos KML generados.
Operaciones	<ul style="list-style-type: none"> • public List<ElementoNombreKML> generar(): genera los documentos KML para el timemap de la administración de incidencias, acciones y reportes. • public String filtrarNombre(String nombre): filtra los nombres de los documentos KML para tratar caracteres especiales. • private String cargarSettingsMapaKML(String tipo, String subtipo, List<Incidencia> listaIncidencias): genera el documento KML para el tipo y subtipo pasados por argumento. • private Element crearArea(Element document, List<PuntosIncidencia> pu, Incidencia s): crea un elemento Element con la información de un área. • private Element crearPunto(Element document, List<PuntosIncidencia> pu, Incidencia i): crea un elemento Element con la información de un punto.

Tabla 141. Clase AdminEventsKML.

	ElementoNombreKML
Responsabilidades	Contiene la información relacionada con la información de un documento KML.
Atributos	<ul style="list-style-type: none"> • private String nombreTipo: nombre del tipo de incidencia, acción o recurso. • private List<String> nombreDocumentos: contiene la lista de documentos KML generados. • private List<String> subtipos: contiene la lista de subtipos del tipo.
Operaciones	Sin operaciones.

Tabla 142. Clase ElementoNombreKML.

4.2...3.2 Clases DAO

En este apartado se explican con detalle todas las clases que implementan los DAO de la aplicación web.

	Incidencia
Responsabilidades	DAO relacionado con una incidencia.
Atributos	<ul style="list-style-type: none"> • private Integer idIncidencia: almacena el identificador de la incidencia. • private Tipo tipo: almacena el tipo de la incidencia. • private Subtipo subtipo: almacena el subtipo de la incidencia. • Private String titulo: almacena el título de la incidencia. • Private String descripción: almacena la descripción de la incidencia. • Private Date fechaInit: almacena la fecha de inicio de la incidencia. • Private Date fechaEnd: almacena la fecha de finalización de la incidencia.
Operaciones	Métodos get y set de cada atributo.

Tabla 143. Clase Incidencia.

	Subincidencia
Responsabilidades	DAO relacionado con una acción.
Atributos	<ul style="list-style-type: none"> • private Integer idSubincidencia: almacena el identificador de la acción. • private Incidencia incidencia: almacena la incidencia a la que pertenece la acción. • private Subtipo subtipo: almacena el subtipo de la acción. • private Tipo tipo: almacena el tipo de la acción. • private String titulo: almacena el título de la acción. • private String descripción: almacena la descripción de la acción. • private Date fechaInit: almacena la fecha de inicio de la acción. • private Date fechaEnd: almacena la fecha de finalización de la acción.
Operaciones	Métodos get y set de cada atributo.

Tabla 144. Clase SubIncidencia.

	Recurso
Responsabilidades	DAO relacionado con un reporte.
Atributos	<ul style="list-style-type: none"> • private Integer idRecurso: almacena el identificador del reporte. • private Incidencia incidencia: almacena la incidencia a la que pertenece el reporte. • private TipoRecurso tipoRecurso: almacena el tipo de reporte. • private String titulo: almacena el título del reporte. • private String longitud: almacena la coordenada longitud del reporte. • private String latitud: almacena la coordenada latitud del reporte. • private String nombreOriginal: almacena el nombre original del reporte. • private String descripción: almacena la descripción del reporte. • private Date fechaInit: almacena la fecha de inicio del reporte. • private Date fechaEnd: almacena la fecha de finalización del reporte.
Operaciones	Métodos get y set de cada atributo.

Tabla 145. Clase Recurso.

	TipoRecurso
Responsabilidades	DAO relacionado con el tipo de reporte.
Atributos	<ul style="list-style-type: none"> • private String tipoRecurso: almacena el tipo de reporte. • private String descripcion: almacena la descripción del tipo de reporte.
Operaciones	Métodos get y set de cada atributo.

Tabla 146. Clase TipoRecurso.

	Tipo
Responsabilidades	DAO relacionado con el tipo de incidencia o acción.
Atributos	<ul style="list-style-type: none"> • private String tipo: almacena el tipo de incidencia o acción. • private String descripcion: almacena la descripción del tipo de incidencia o acción.
Operaciones	Métodos get y set de cada atributo.

Tabla 147. Clase Tipo.

	SubTipo
Responsabilidades	DAO relacionado con el subtipo de incidencia o acción.
Atributos	<ul style="list-style-type: none"> • private String subtipo: almacena el tipo de incidencia o acción.

	<ul style="list-style-type: none"> • private String tipo: almacena el tipo al que pertenece el subtipo. • private String descripcion: almacena la descripción del subtipo de incidencia o acción.
Operaciones	Métodos get y set de cada atributo.

Tabla 148. Clase SubTipo.

	PuntosIncidencia
Responsabilidades	DAO relacionado con la ubicación de una incidencia.
Atributos	<ul style="list-style-type: none"> • private Integer idPuntosIncidencia: almacena el identificador del punto. • private Incidencia incidencia: almacena la incidencia a la que pertenece el punto. • private String longitud: almacena la longitud del punto. • private String latitud: almacena la latitud del punto. • private int posicion: almacena la posición del punto.
Operaciones	Métodos get y set de cada atributo.

Tabla 149. Clase PuntosIncidencia.

	PuntosSubincidencia
Responsabilidades	DAO relacionado con la ubicación de una acción.
Atributos	<ul style="list-style-type: none"> • private Integer idPuntosSubincidencia: almacena el identificador del punto. • private Subincidencia subincidencia: almacena la acción a la que pertenece el punto. • private String longitud: almacena la longitud del punto. • private String latitud: almacena la latitud del punto. • private int posicion: almacena la posición del punto.
Operaciones	Métodos get y set de cada atributo.

Tabla 150. Clase PuntosSubincidencia.

	Usuarios
Responsabilidades	DAO relacionado con la información de un usuario.
Atributos	<ul style="list-style-type: none"> • private String usuario: almacena el identificador del punto. • private Perfil perfil: almacena el perfil del usuario. • private String password: almacena la contraseña del usuario. • private String nombre: almacena el nombre del usuario. • private String apellidos: almacena los apellidos del usuario. • private String teléfono: almacena el número de teléfono del usuario. • private String email: almacena el e-mail del usuario.
Operaciones	Métodos get y set de cada atributo.

Tabla 151. Clase Usuarios.

	Perfil
Responsabilidades	DAO relacionado con la información del perfil de un usuario.
Atributos	<ul style="list-style-type: none"> • private String usuario: almacena el nombre del perfil. • private String descripcion: almacena la descripción del perfil.
Operaciones	Métodos get y set de cada atributo.

Tabla 152. Clase Perfil.

4.2...3.3 Beans

En esta sección se explicarán con detalle todas las clases que se comportan como beans de la aplicación. Estos beans serán utilizados por JSF desde la capa vista.

	UserBean
Responsabilidades	Bean relacionado con la información de logeo de un usuario.
Atributos	<ul style="list-style-type: none"> • private String user: almacena el usuario de logeo. • private String password: almacena la contraseña de logeo.
Operaciones	<ul style="list-style-type: none"> • public String login(): se encarga realizar el logeo el usuario.

Tabla 153. Clase UserBean.

	SubmitEventBean
Responsabilidades	Bean relacionado con la información de creación de una incidencia.
Atributos	<ul style="list-style-type: none"> • private String title_event: almacena el título de la incidencia. • private String type_event: almacena el tipo de la incidencia. • private String subtype_event: almacena el subtipo de la incidencia. • private String start_date_event: almacena la fecha de inicio de la incidencia. • private String end_date_event: almacena la fecha de finalización de la incidencia. • private String description_event: almacena la descripción de la incidencia. • private String points_hidden: almacena los puntos de la ubicación de la incidencia.
Operaciones	<ul style="list-style-type: none"> • public String save_event(): se encarga de guardar la incidencia. • Public List<SelectItem> getSelectItemsTipo(): se encarga de enviar los tipos de incidencia a la página JSP. • Public List<SelectItem> setSelectItemsSubTipo(): se encarga de enviar los subtipos de incidencia a la página JSP.

Tabla 154. Clase SubmitEventBean.

	EditEventBean
Responsabilidades	Bean relacionado con la información de edición de una incidencia.
Atributos	<ul style="list-style-type: none"> • private String title_event: almacena el título de la incidencia. • private String type_event: almacena el tipo de la incidencia. • private String subtype_event: almacena el subtipo de la incidencia. • private String start_date_event: almacena la fecha de inicio de la incidencia. • private String end_date_event: almacena la fecha de finalización de la incidencia. • private String description_event: almacena la descripción de la incidencia. • private String conservar_ubicacion_hidden: almacena si se debe conservar

	<p>la ubicación o recoger la recibida.</p> <ul style="list-style-type: none"> • private String points_hidden: almacena los puntos de la ubicación de la incidencia. • private String id_event: almacena el identificador original del evento.
Operaciones	<ul style="list-style-type: none"> • public String update_event(): se encarga de actualizar la incidencia. • Public List<SelectItem> getSelectItemsTipo(): se encarga de enviar los tipos de incidencia a la página JSP. • Public List<SelectItem> setSelectItemsSubTipo(): se encarga de enviar los subtipos de incidencia a la página JSP.

Tabla 155. Clase EditEventBean.

SubmitSubEventBean	
Responsabilidades	Bean relacionado con la información de creación de una acción.
Atributos	<ul style="list-style-type: none"> • private String title_event: almacena el título de la acción. • private String type_event: almacena el tipo de la acción. • private String subtype_event: almacena el subtipo de la acción. • private String start_date_event: almacena la fecha de inicio de la acción. • private String end_date_event: almacena la fecha de finalización de la acción. • private String description_event: almacena la descripción de la acción. • private String points_hidden: almacena los puntos de la ubicación de la acción. • private String id_main_event: almacena el identificador de la incidencia a la que pertenece la acción.
Operaciones	<ul style="list-style-type: none"> • public String save_subevent(): se encarga de guardar la acción. • Public List<SelectItem> getSelectItemsTipo(): se encarga de enviar los tipos de incidencia a la página JSP. • Public List<SelectItem> setSelectItemsSubTipo(): se encarga de enviar los subtipos de incidencia a la página JSP.

Tabla 156. Clase SubmitSubEventBean.

EditSubEventBean	
Responsabilidades	Bean relacionado con la información de edición de una acción.
Atributos	<ul style="list-style-type: none"> • private String title_event: almacena el título de la acción. • private String type_event: almacena el tipo de la acción. • private String subtype_event: almacena el subtipo de la acción. • private String start_date_event: almacena la fecha de inicio de la acción. • private String end_date_event: almacena la fecha de finalización de la acción. • private String description_event: almacena la descripción de la acción. • private String conservar_ubicacion_hidden: almacena si se debe conservar la ubicación almacenada o se debe reemplazar por la recibida. • private String points_hidden: almacena los puntos de la ubicación de la acción. • private String id_subevent: almacena el identificador de la acción.
Operaciones	<ul style="list-style-type: none"> • public String update_subevent(): se encarga de actualizar la acción. • Public List<SelectItem> getSelectItemsTipo(): se encarga de enviar los tipos de incidencia a la página JSP. • Public List<SelectItem> setSelectItemsSubTipo(): se encarga de enviar los subtipos de incidencia a la página JSP.

Tabla 157. Clase EditSubEventBean.

	SubmitResourceBean
Responsabilidades	Bean relacionado con la información de creación de un reporte.
Atributos	<ul style="list-style-type: none"> • private String title_resource: almacena el título del reporte. • private String type_resource: almacena el tipo del reporte. • private String start_date_resource: almacena la fecha de inicio del reporte. • private String description_resource: almacena la descripción del reporte. • private String points_hidden: almacena los puntos de la ubicación del reporte. • private UploadedFile uploadedFile: almacena el archivo subido. • private String fileName: almacena el nombre del archivo subido. • private String id_main_event: almacena el identificador de la incidencia a la que pertenece el reporte.
Operaciones	<ul style="list-style-type: none"> • public String save_resource(): se encarga de guardar el reporte.

Tabla 158. Clase SubmitResourceBean.

	EditResourceBean
Responsabilidades	Bean relacionado con la información de edición de un reporte.
Atributos	<ul style="list-style-type: none"> • private String title_resource: almacena el título del reporte. • private String type_resource: almacena el tipo del reporte. • private String start_date_resource: almacena la fecha de inicio del reporte. • private String description_resource: almacena la descripción del reporte. • private String conservar_ubicacion_hidden: indica si se debe mantener la ubicación guardada o se tiene que sustituir por la ubicación enviada por el usuario. • private String points_hidden: almacena los puntos de la ubicación del reporte. • private String conservar_file_hidden: indica si se debe mantener el archivo guardada o se tiene que sustituir por el archivo enviado por el usuario. • private UploadedFile uploadedFile: almacena el archivo subido. • private String fileName: almacena el nombre del archivo subido. • private String id_resource: almacena el identificador del reporte que se tiene que actualizar.
Operaciones	<ul style="list-style-type: none"> • public String edit_resource(): se encarga de guardar el reporte. • public List<SelectItem> getSelectItemsTipo(): envía al JSP la lista de tipos de reportes.

Tabla 159. Clase EditResourceBean.

	SubmitUserBean
Responsabilidades	Bean relacionado con la información de edición y creación de un usuario.
Atributos	<ul style="list-style-type: none"> • private String usuario: almacena el nombre de usuario del usuario. • private String nombre: almacena el nombre del usuario. • private String apellidos: almacena los apellidos del usuario. • private String telefono: almacena el teléfono del usuario. • private String email: almacena el e-mail del usuario. • private String perfil: almacena el perfil del usuario. • private String contraseña: almacena la contraseña del usuario. • private String anterior_usuario: almacena identificador del usuario original

	<p>en el caso de que se quiera cambiar.</p> <ul style="list-style-type: none"> • private String sabe_new_pass: indica si se debe cambiar la contraseña o mantener la almacenada en el sistema.
Operaciones	<ul style="list-style-type: none"> • public String edit_user(): se encarga de editar el usuario. • Public String save_user(): se encarga de guardar el usuario. • public List<SelectItem> getSelectItemsTipo(): envía al JSP la lista de tipos de usuario.

Tabla 160. Clase SubmitUserBean.

4.2...4 Capa infraestructura

La capa infraestructura del sistema se encarga de la comunicación con la base de datos de la aplicación. Para ello se utilizará un la tecnología Hibernate, que permite poder migrar fácilmente de un tipo de base de datos a otro.

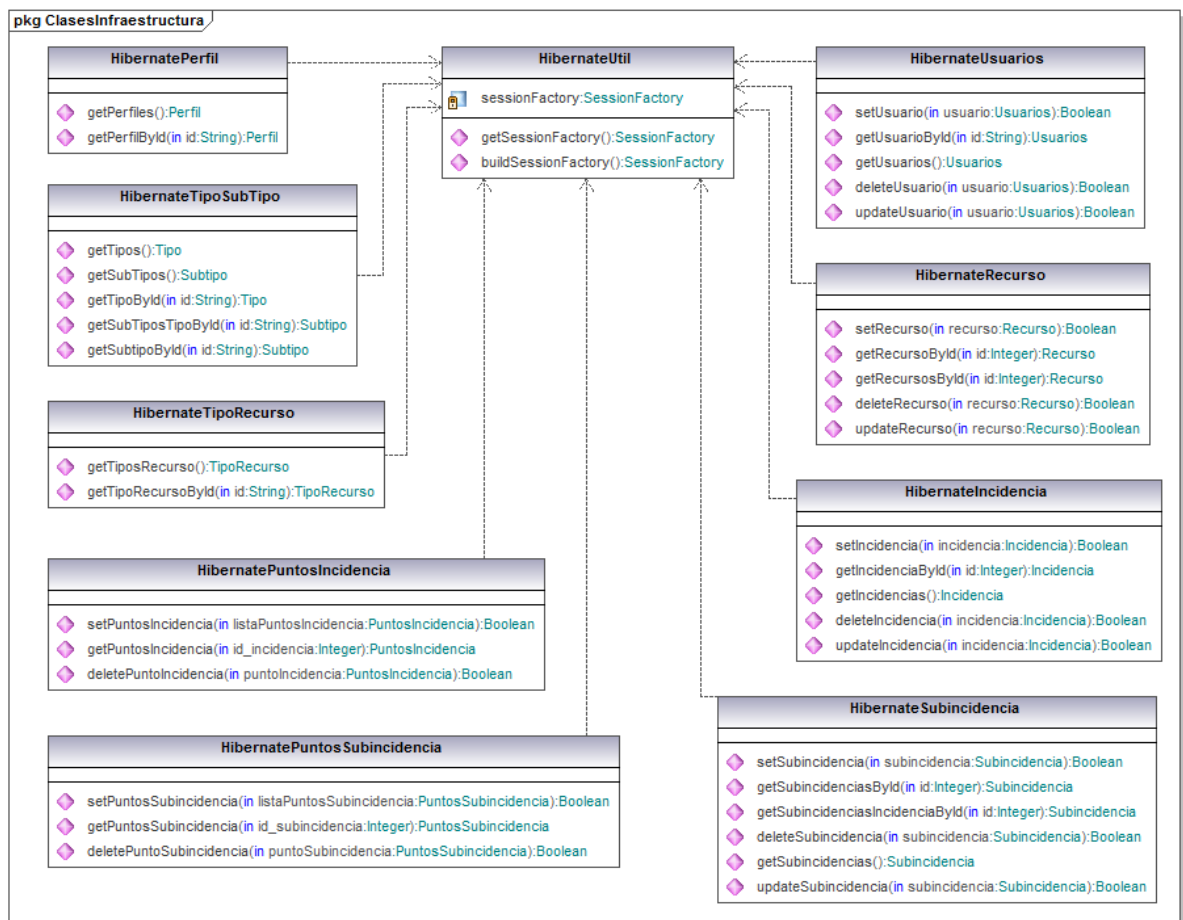


Figura 21. Modelo de clases de la capa Infraestructura.

En la Figura 21. Modelo de clases de la capa Infraestructura. aparecen todas las clases necesarias para la comunicación del sistema con la base de datos utilizando la tecnología hibernate.

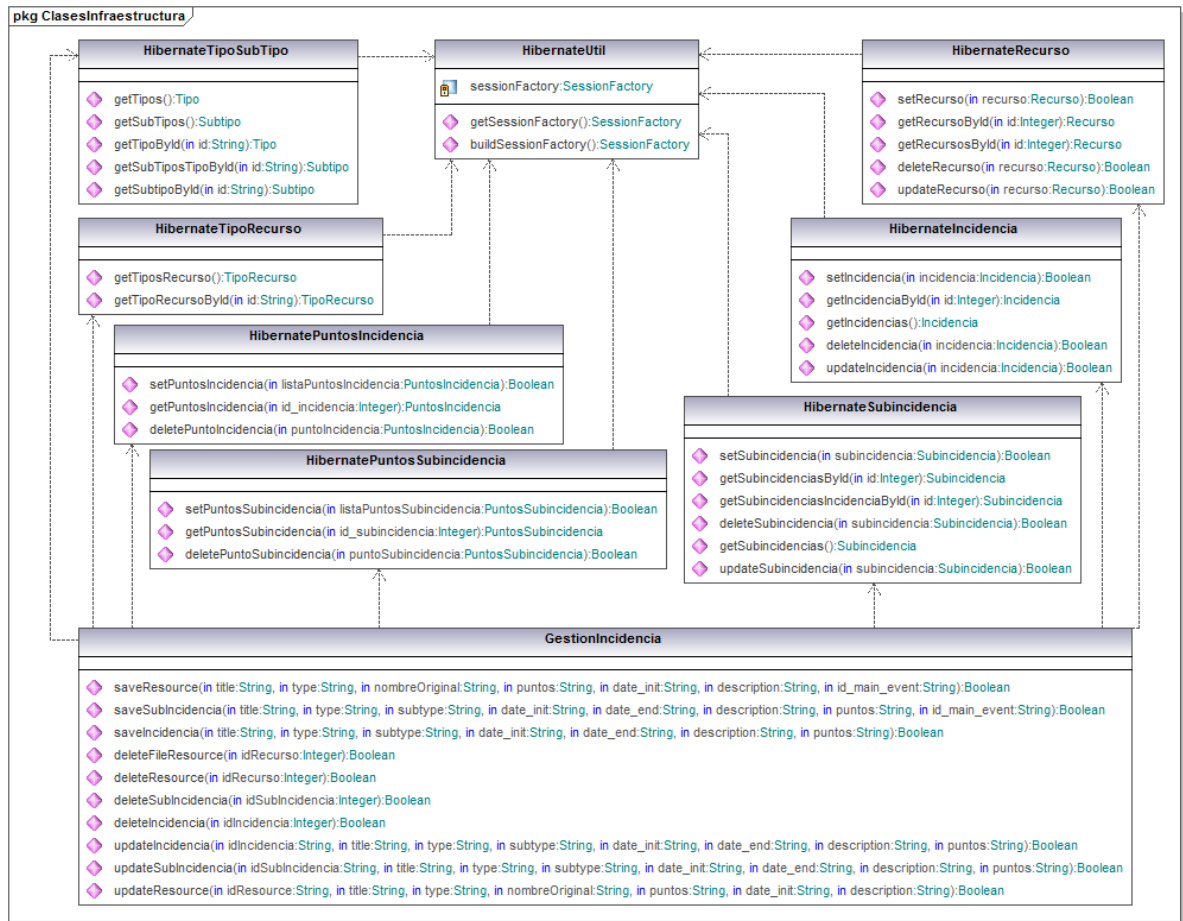


Figura 22. Modelo de clases de la gestión de incidencias con la capa Infraestructura.

En la Figura 22. Modelo de clases de la gestión de incidencias con la capa Infraestructura. se muestra la relación entre la gestión de incidencias de la capa *Modelo* entre la capa *Infraestructura*.

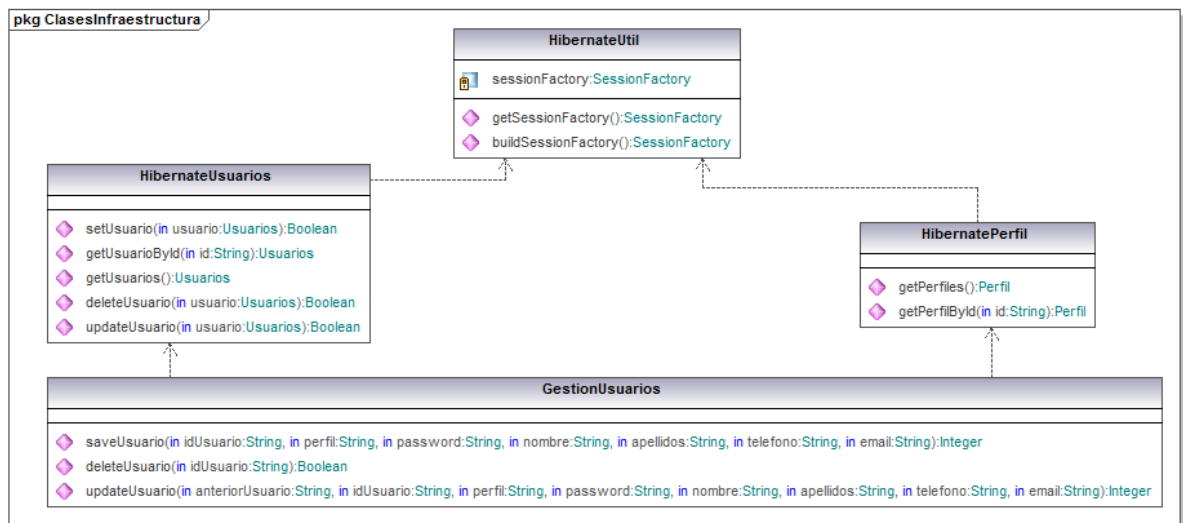


Figura 23. Modelo de clases de la gestión de usuarios con la capa Infraestructura.

En la Figura 23. Modelo de clases de la gestión de usuarios con la capa Infraestructura. se muestra la relación entre el conjunto de clases encargada de la gestión de usuarios (capa *Modelo*) y la capa *Infraestructura*.

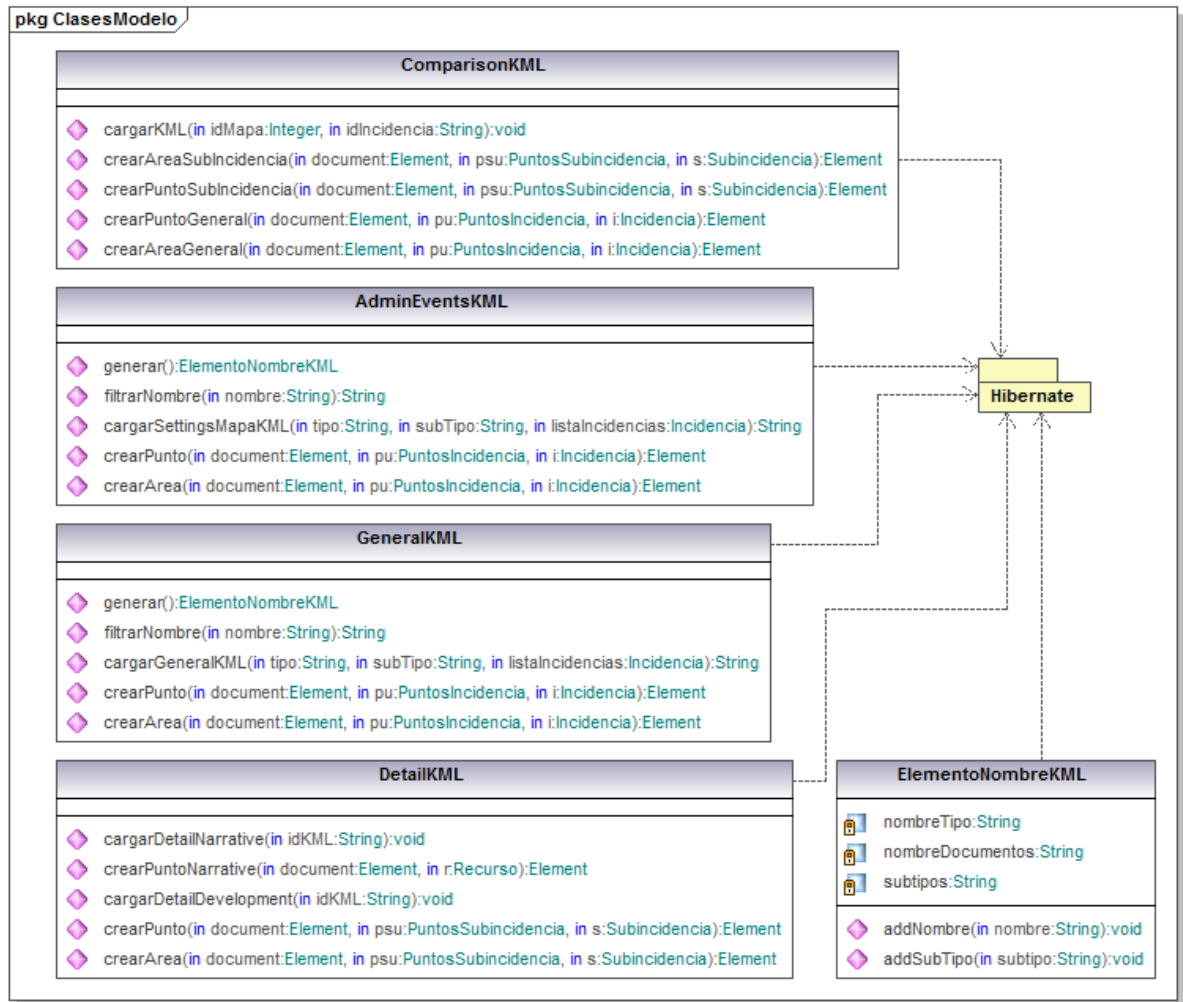


Figura 24. Modelo de clases de la gestión de documentos KML con la capa Infraestructura.

En la Figura 24. Modelo de clases de la gestión de documentos KML con la capa Infraestructura. se muestra la relación entre el conjunto de clases encargadas con la gestión de documentos KML y la capa Infraestructura. En este caso la capa infraestructura se representa mediante el paquete *Hibernate*.

A continuación se describirán con detalle todas las clases de la capa *Infraestructura*.

	HibernateUtil
Responsabilidades	Almacena la factoría de sesiones de hibernate.
Atributos	<ul style="list-style-type: none"> private static final SessionFactory sessionFactory: almacena un objeto estática de tipo SessionFactory. Se encarga de almacenar las sesiones que el sistema puede tener activas.
Operaciones	<ul style="list-style-type: none"> public static SessionFactory buildSessionFactory(SessionFactory sessionFactory): se encarga de instanciar el objeto sessionFactory y pasarle

la configuración de inicio.

Tabla 161. Clase HibernateUtil.

	HibernateUsuarios
Responsabilidades	Contiene las operaciones de acceso a los datos de usuarios.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean setUsuario(Usuarios usuario): se encarga de guardar un usuario en la base de datos. • public Usuarios getUsuarioById(String id): se encarga de recuperar un usuario de la base de datos con el identificador pasado por argumento. • public List<Usuarios> getUsuarios(): devuelve una lista con todos los usuarios de la base de datos. • public boolean deleteUsuario(Usuarios usuario): elimina de la base de datos el usuario pasado por argumento. • Public boolean updateUsuario(Usuarios usuario): actualiza el usuario pasado por argumento.

Tabla 162. Clase HibernateUsuarios.

	HibernatePerfil
Responsabilidades	Contiene las operaciones de acceso a los datos de perfiles de usuario.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public Usuarios getPerfilesById(String id): se encarga de recuperar un perfil de la base de datos con el identificador pasado por argumento. • public List<Usuarios> getPerfiles(): devuelve una lista con todos los perfiles de usuarios de la base de datos.

Tabla 163. Clase HibernatePerfil.

	HibernateIncidencia
Responsabilidades	Contiene las operaciones de acceso a los datos de usuarios.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean setIncidencia(Incidencia incidencia): se encarga de guardar una incidencia en la base de datos. • public Incidencia getIncidenciaById(int id): se encarga de recuperar una incidencia de la base de datos con el identificador pasado por argumento. • public List<Incidencia> getIncidencias(): devuelve una lista con todas las incidencias de la base de datos. • public boolean deleteIncidencia(Incidencia incidencia): elimina de la base de datos la incidencia pasada por argumento. • public boolean updateIncidencia(Incidencia incidencia): actualiza la incidencia pasada por argumento.

Tabla 164. Clase HibernateIncidencia.

	HibernateSubIncidencia
Responsabilidades	Contiene las operaciones de acceso a los datos de usuarios.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean setSubIncidencia(Subincidencia subincidencia): se encarga de guardar una subincidencia en la base de datos. • public Subincidencia getSubincidenciaById(int id): se encarga de recuperar una subincidencia de la base de datos con el identificador pasado por

	<p>argumento.</p> <ul style="list-style-type: none"> • public List<Subincidencia> getSubincidenciasIncidenciaById(int id): devuelve una lista con todas las subincidencias de una incidencia de la base de datos. • public boolean deleteSubIncidencia(Subincidencia subincidencia): elimina de la base de datos la subincidencia pasada por argumento. • public boolean updateSubincidencia(Subincidencia subincidencia): actualiza la subincidencia pasada por argumento. • public List<Subincidencias> getSubincidencias(): devuelve una lista con todas las subincidencias de la base de datos.
--	---

Tabla 165. Clase HibernateSubIncidencia.

	HibernateRecurso
Responsabilidades	Contiene las operaciones de acceso a los datos de un reporte.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean setRecurso(Recurso recurso): se encarga de guardar un recurso en la base de datos. • public Recurso getRecursoById(int id): se encarga de recuperar un recurso de la base de datos con el identificador pasado por argumento. • public List<Recurso> getRecursosById(int id): devuelve una lista con todos los recursos de una incidencia de la base de datos. • public boolean deleteRecurso(Recurso recurso): elimina de la base de datos el recurso pasado por argumento. • public boolean updateRecurso(Recurso recurso): actualiza el recurso pasado por argumento.

Tabla 166. Clase HibernateRecurso.

	HibernateTipoSubTipo
Responsabilidades	Contiene las operaciones de acceso a los datos tipos y subtipos de incidencias y acciones.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public List<Tipo> getTipos(): devuelve una lista con todos los tipos de incidencias o acciones de la base de datos. • public List<Subtipo> getSubTipos(): devuelve una lista con todos los subtipos de incidencias o acciones de la base de datos. • public Tipo getTipoById(String id): devuelve el tipo que tiene el identificador pasado por argumento. • public List<Subtipo> getSubTiposTipoById(String id): devuelve una lista con todos los subtipos de un tipo de la base de datos. • public Tipo getSubtipoById(String id): devuelve el subtipo que tiene el identificador pasado por argumento.

Tabla 167. Clase HibernateTipoSubTipo.

	HibernateTipoRecurso
Responsabilidades	Contiene las operaciones de acceso a los datos de tipos de recurso de un reporte.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public List<TipoRecurso> getTiposRecurso(): devuelve una lista con todos los tipos de reportes de la base de datos. • public TipoRecurso getTipoRecursoById(String id): devuelve el tipo de

	reporte que tiene el identificador pasado por argumento.
--	--

Tabla 168. Clase HibernateTipoRecurso.

	HibernateRecurso
Responsabilidades	Contiene las operaciones de acceso a los datos ubicación de una incidencia.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean setPuntosIncidencia(List<PuntosIncidencia> listaPuntosIncidencia): se encarga de guardar los puntos de la ubicación de una incidencia en la base de datos. • public List<PuntosIncidencia> getPuntosIncidencia(int id_incidencia): devuelve una lista con todos los puntos de una incidencia de la base de datos. • public boolean deletePuntoIncidencia(PuntoIncidencia puntoIncidencia): elimina de la base de datos el punto pasado por argumento.

Tabla 169. Clase HibernatePuntosIncidencia.

	HibernatePuntosSubIncidencia
Responsabilidades	Contiene las operaciones de acceso a los datos ubicación de una acción de una incidencia.
Atributos	Sin atributos.
Operaciones	<ul style="list-style-type: none"> • public boolean setPuntosSubincidencia(List<PuntosSubincidencia> listaPuntosSubincidencia): se encarga de guardar los puntos de la ubicación de una acción de una incidencia en la base de datos. • public List<PuntosSubincidencia> getPuntosSubincidencia(int id_subincidencia): devuelve una lista con todos los puntos de una acción de una incidencia de la base de datos. • public boolean deletePuntoSubincidencia(PuntoSubincidencia puntoSubincidencia): elimina de la base de datos el punto pasado por argumento.

Tabla 170. Clase HibernatePuntosSubIncidencia.

4.2...5 Modelo lógico de datos

En este apartado se define el modelo de datos que será instaurado en la base de datos y que utilizará el sistema que se está diseñando.

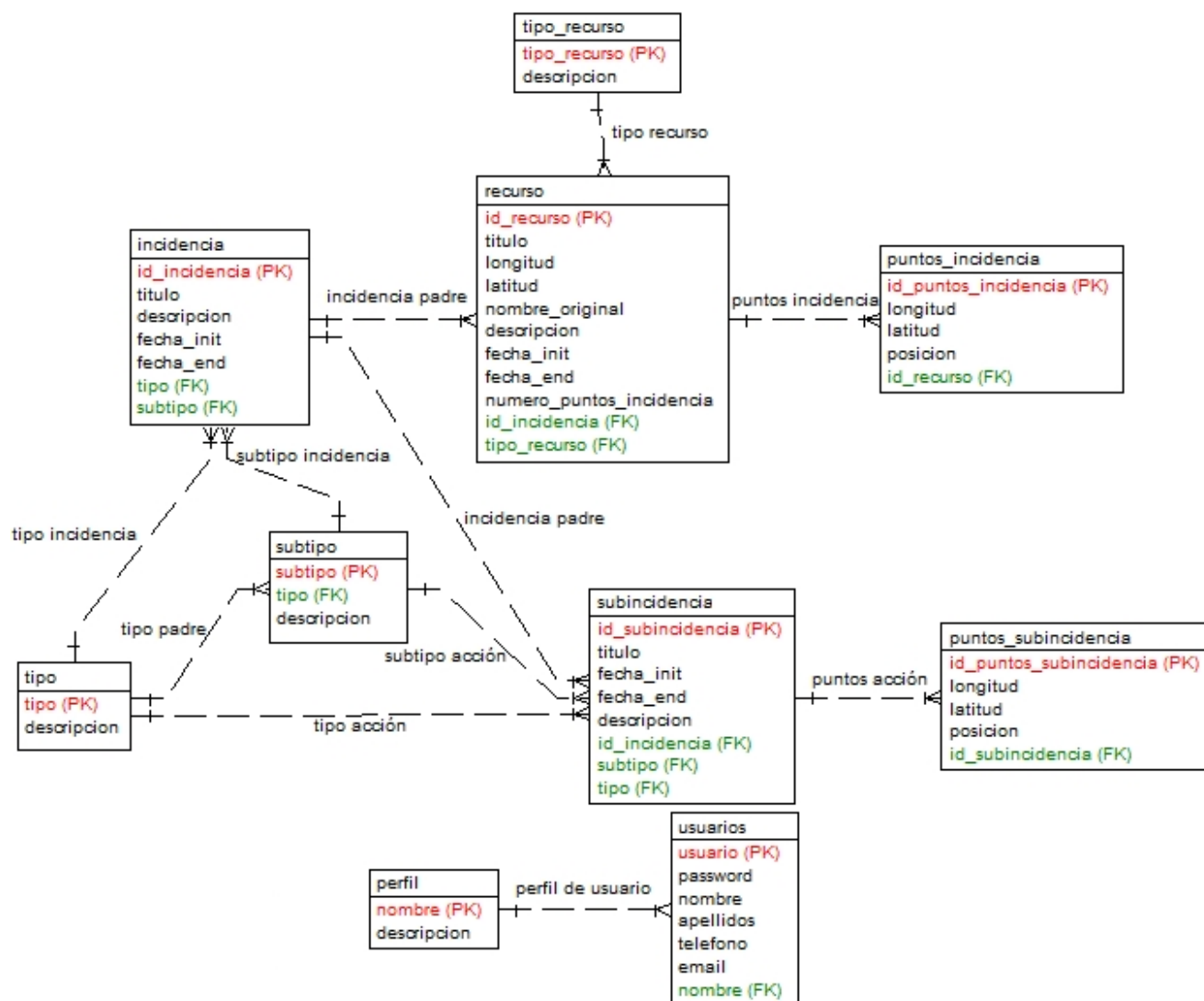


Figura 25. Modelo de datos.

En el anterior modelo se puede observar que el sistema almacenará cada incendio forestal en la tabla incidencia. Cada incidencia tendrá de cero a n reportes (tabla recurso), de cero a n acciones (tabla subincidencia). Cada incidencia será de un tipo y un subtipo, al igual que cada acción. Cada reporte (recurso) tendrá un tipo asociado que indicará el tipo de reporte que es.

La ubicación de las incidencias se almacenará en la tabla puntos_incidencia. La ubicación de cada reporte se guardará en el propio reporte, ya que estará asociado a un único punto. Y la ubicación de cada acción se guardará en la tabla puntos_subincidencia.

La información de cada usuario se almacenará en la tabla usuarios, y cada uno tendrá asociado un perfil.

Implementación

En este apartado se indicará que herramientas se han utilizado en el desarrollo del sistema y que tipo de organización se ha seguido.

Herramientas utilizadas

En este apartado se detallan todas las herramientas que se han utilizado en el desarrollo de la aplicación web.

Herramienta	Especificaciones
Sistema Operativo	Windows 7
IDE	Eclipse Helios Java EE IDE.
Java	Java SE 1.6
Acceso a base de datos	Hibernate 3.0
Conector MySQL	MySQL Conector 5.1.13
Lector XML	Dom4j 1.6.1
Ant	Ant 2.7.6
Gestión conexiones	C3P0 0.9.1
Captura de errores	Log4j
Timemap	Timemap 1.6
GoogleMaps	GoogleMaps 2.0
JSF	JSF 1.2
JSP	Dynamic Web Module 2.5
Base de datos	MySQL 5.5
Servidor	Apache Tomcat 7.0
Diagramas y Modelos	Altova UModel 2011
Diagramas y Modelos	Case Studio 2

Tabla 171. Herramientas utilizadas durante el desarrollo.

Organización del proyecto

Se ha utilizado una organización por carpetas en las que se indica el contenido y la fecha. Cada archivo creado contiene una versión probada y estable del sistema desarrollado.

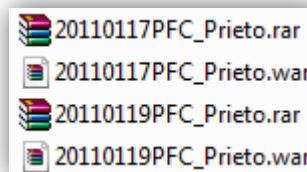


Figura 26. Organización del proyecto.

En la anterior figura, cada archivo *.rar* contiene el proyecto Eclipse y cada elemento *.war* es el archivo que se tiene que desplegar en el servidor. En el nombre se indica la fecha en el formato "aaaammdd", de esta manera siempre aparecerán todos los archivos ordenados en orden creciente según la fecha.

Se ha utilizado una lista de elementos para llevar un control sobre las tareas pendientes y realizadas tal y como se puede ver en la siguiente figura:

```

Controlar el tamaño de cada campo, sino dará error - ARREGLADO
En submit_subevent recuperar el idMainEvent - ARREGLADO
Fechas solo se cambian con javascript y no el usuario - ARREGLADO
Al añadir un subevento redireccionar a detail del evento padre - ARREGLADO
Imagen del timemap que no carga - ARREGLADO
Poner tipos de recurso - ARREGLADO
Descargar recursos - ARREGLADO
Pinchar recurso y seleccionar en el mapa en DETAIL.JSF - ARREGLADO
Barra de progreso de subida de archivo - ARREGLADO
Poner tipos de incidencia - ARREGLADO
Borrar el archivo del recurso de la carpeta /resource - ARREGLADO
Autoseleccionar fecha actual en submit event, subevent y resource - ARREGLADO
Pinchar subevento y seleccionar en el mapa en DETAIL.JSF - ARREGLADO
Añadir un icono para pulsar en un subevento de DETAIL.JSF - ARREGLADO
Borrar recursos sin File asociado, pero informar en el log - ARREGLADO
Añadir icono de refresco de mapa - ARREGLADO
Filtros por tipo y subtipo - ARREGLADO
Resaltar el menú en el que se está - ARREGLADO
Codigo timemap de DETAIL.JSP al 1.6 - ARREGLADO
Mensajes del log al borrar incidencia - ARREGLADO
Mensajes del log al crear KML's, hay muchos - ARREGLADO
Modo texto en GENERAL.JSP - ARREGLADO
Administración de incidencias borrado - ARREGLADO
Login de usuario - ARREGLADO
Administración de incidencias edición modo lista - ARREGLADO
Administración de usuarios - ARREGLADO

```

Figura 27. Lista de tareas del desarrollo.

Pruebas

En esta sección se incluirán todos los informes de las pruebas unitarias e integración. De esta forma se podrán validar los requisitos y comprobar el correcto funcionamiento del sistema construido. La definición de estas pruebas y los requisitos para poder llevarlas a cabo está definida en el apartado **Plan de pruebas**.

Primero se debe preparar el entorno de ejecución, disponiendo de todos los recursos necesarios para realizar las pruebas unitarias de cada uno de los componentes del sistema de información. Se debe asegurar la disponibilidad del entorno y de los datos necesarios, bibliotecas o librerías oportunas, así como los procedimientos manuales o automáticos necesarios.

Pruebas de Caja Blanca

Las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Se escogerán distintos valores de entrada para cada procedimiento y cerciorarse de que se devuelven los valores de salida adecuados.

PB-01-DAO			
Objetivo	Devuelve un objeto SessionFactory.		
Componente	DAO	Clase	HibernateUtil
Método	getSessionFactory		
Entrada	void		
Salida	Objeto de tipo SessionFactory.		
Precondiciones	Debe existir el archivo hibernate.cfg.xml.		

	Debe existir la base de datos y ser accesible.
Pseudocódigo	return sessionFactory;
Secuencia de ejecución válida	Se invoca el método. Se comprueba que se puede operar con el objeto.

Tabla 172. PB-01-DAO.

PB-02-DAO			
Objetivo	Insertar valores a un objeto de tipo Incidencia.		
Componente	DAO	Clase	Incidencia
Método	setters		
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Salida	void		
Precondiciones	Parámetros de entrada distintos de null.		
Pseudocódigo	<pre> Incidencia incidencia = new Incidencia (); incidencia.titulo = titulo; incidencia.tipo = tipo; incidencia.subtipo = subtipo; incidencia.fechaIni = fechaIni; incidencia.fechaFin = fechaFin; incidencia.puntos = puntos incidencia.descripcion = descripción; //Imprimir cada uno de los valores introducidos System.out.println(incidencia); </pre>		
Secuencia de ejecución válida	Se invoca el método pasándole como argumento los parámetros de la incidencia. Se comprueba que se imprime por pantalla los valores que se pasaron.		

Tabla 173. PB-02-DAO.

PB-03-DAO			
Objetivo	Insertar valores a un objeto de tipo Subincidencia.		
Componente	DAO	Clase	SubIncidencia
Método	setters		
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Salida	void		
Precondiciones	Parámetros de entrada distintos de null.		
Pseudocódigo	<pre> Subincidencia subincidencia = new Subincidencia(); subincidencia.titulo = titulo; subincidencia.tipo = tipo; subincidencia.subtipo = subtipo; subincidencia.fechaIni = fechaIni; subincidencia.fechaFin = fechaFin; subincidencia.puntos = puntos </pre>		

	subincidencia.descripcion = descripción; subincidencia.idPadre = idPadre; //Imprimir cada uno de los valores introducidos System.out.println(subincidencia);
Secuencia de ejecución válida	Se invoca el método pasándole como argumento los parámetros de la acción. Se comprueba que se imprime por pantalla los valores que se pasaron.

Tabla 174. PB-03-DAO.

PB-04-DAO			
Objetivo	Insertar valores a un objeto de tipo Recurso.		
Componente	DAO	Clase	Recurso
Método	setters		
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Salida	void		
Precondiciones	Parámetros de entrada distintos de null.		
Pseudocódigo	Recurso recurso = new Recurso(); recurso.titulo = titulo; recurso.nombreOriginal = nombreOriginal; recurso.puntos = puntos; recurso.fecha = fecha recurso.descripcion = descripción; recurso.idPadre = idPadre; //Imprimir cada uno de los valores introducidos System.out.println(recurso);		
Secuencia de ejecución válida	Se invoca el método pasándole como argumento los parámetros del reporte. Se comprueba que se imprime por pantalla los valores que se pasaron.		

Tabla 175. PB-04-DAO.

PB-05-DAO			
Objetivo	Insertar valores a un objeto de tipo Usuario.		
Componente	DAO	Clase	Usuario
Método	setters		
Entrada	Nick, perfil, contraseña, nombre, apellidos, teléfono y e-mail.		
Salida	void		
Precondiciones	Parámetros de entrada distintos de null.		
Pseudocódigo	Usuario usuario = new Usuario(); usuario.nick = nick; usuario.contraseña = contraseña; usuario.nombre = nombre;		

	usuario.apellidos = apellidos usuario.telefono = telefono; usuario.email = email; //Imprimir cada uno de los valores introducidos System.out.println(usuario);
Secuencia de ejecución válida	Se invoca el método pasándole como argumento los parámetros del usuario. Se comprueba que se imprime por pantalla los valores que se pasaron.

Tabla 176. PB-05-DAO.

PB-06-DAO			
Objetivo	Obtener valores de un objeto de tipo Incidencia.		
Componente	DAO	Clase	Incidencia
Método	getters		
Entrada	void		
Salida	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Precondiciones			
Pseudocódigo	System.out.println(incidencia.getTitulo()); System.out.println(incidencia.getTipo()); System.out.println(incidencia.getSubtipo()); System.out.println(incidencia.getFechaIni()); System.out.println(incidencia.getFechaFin()); System.out.println(incidencia.getPuntos()); System.out.println(incidencia.getDescripcion());		
Secuencia de ejecución válida	Se invoca el método. Se comprueba que se imprime por pantalla los valores esperados.		

Tabla 177. PB-06-DAO.

PB-07-DAO			
Objetivo	Obtener valores de un objeto de tipo Subincidencia.		
Componente	DAO	Clase	SubIncidencia
Método	getters		
Entrada	void		
Salida	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Precondiciones			
Pseudocódigo	System.out.println(subincidencia.getTitulo()); System.out.println(subincidencia.getTipo()); System.out.println(subincidencia.getSubtipo()); System.out.println(subincidencia.getFechaIni()); System.out.println(subincidencia.getFechaFin()); System.out.println(subincidencia.getPuntos()); System.out.println(subincidencia.getDescripcion());		

Secuencia de ejecución válida	Se invoca el método. Se comprueba que se imprime por pantalla los valores que se pasaron.
--------------------------------------	--

Tabla 178. PB-07-DAO.

PB-08-DAO			
Objetivo	Obtener valores de un objeto de tipo Recurso.		
Componente	DAO	Clase	Recurso
Método	getters		
Entrada	void		
Salida	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Precondiciones			
Pseudocódigo	System.out.println(recurso.getTitulo()); System.out.println(recurso.getTipo()); System.out.println(recurso.getFechaIni()); System.out.println(recurso.getPuntos()); System.out.println(recurso.getDescripcion());		
Secuencia de ejecución válida	Se invoca el método. Se comprueba que se imprime por pantalla los valores que se pasaron.		

Tabla 179. PB-08-DAO.

PB-09-DAO			
Objetivo	Obtener valores de un objeto de tipo Usuario.		
Componente	DAO	Clase	Usuario
Método	getters		
Entrada	void		
Salida	Nick, perfil, contraseña, nombre, apellidos, teléfono y e-mail.		
Precondiciones			
Pseudocódigo	System.out.println(usuario.getNick()); System.out.println(usuario.getPerfil()); System.out.println(usuario.getContraseña()); System.out.println(usuario.getNombre()); System.out.println(usuario.getApellidos()); System.out.println(usuario.getTelefono()); System.out.println(usuario.getEmail());		
Secuencia de ejecución válida	Se invoca el método. Se comprueba que se imprime por pantalla los valores que se pasaron.		

Tabla 180. PB-09-DAO.

Pruebas de Caja Negra

Pruebas de caja negra: se centra en pruebas en las que no se tiene en cuenta el código, sino la respuesta a una serie de entradas. En otras palabras, estas pruebas se centran en comprobar la

forma de interactuar con el medio teniendo en cuenta que debe hacer pero no como debe hacerlo.

PN-01-DAO			
Objetivo	Devuelve un objeto de tipo Incidencia.		
Componente	DAO	Clase	Incidencia
Método	getIncidenciaById		
Entrada	Número entero que representa la clave primaria del objeto.		
Salida	Objeto de tipo Incidencia.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible. Debe existir una incidencia con la clave primaria introducida.		
Pseudocódigo	Incidencia incidencia = session.uniqueResult(); return incidencia;		
Secuencia de ejecución válida	Se invoca el método pasándole como argumento la clave primaria de la incidencia que se quiere buscar. Se comprueba que el objeto devuelto corresponde con el esperado.		

Tabla 181. PN-01-DAO.

PN-02-DAO			
Objetivo	Devuelve un objeto de tipo Subincidencia (Acción).		
Componente	DAO	Clase	Subincidencia
Método	getSubincidenciaById		
Entrada	Número entero que representa la clave primaria del objeto.		
Salida	Objeto de tipo Subincidencia.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible. Debe existir una acción con la clave primaria introducida.		
Pseudocódigo	Subincidencia subincidencia = session.uniqueResult(); return subincidencia;		
Secuencia de ejecución válida	Se invoca el método pasándole como argumento la clave primaria de la acción que se quiere buscar. Se comprueba que el objeto devuelto corresponde con el esperado.		

Tabla 182. PN-02-DAO.

PN-03-DAO			
Objetivo	Devuelve un objeto de tipo Recurso (Reporte).		
Componente	DAO	Clase	Recurso
Método	getRecursoById		
Entrada	Número entero que representa la clave primaria del objeto.		
Salida	Objeto de tipo Recurso.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible. Debe existir un reporte con la clave primaria introducida.		
Pseudocódigo	Recurso recurso = session.uniqueResult(); return recurso;		

Secuencia de ejecución válida	Se invoca el método pasándole como argumento la clave primaria del reporte que se quiere buscar. Se comprueba que el objeto devuelto corresponde con el esperado.
--------------------------------------	--

Tabla 183. PB-03-DAO.

PN-04-DAO			
Objetivo	Devuelve un objeto de tipo Usuario.		
Componente	DAO	Clase	Usuario
Método	getUsuarioById		
Entrada	Número entero que representa la clave primaria del objeto.		
Salida	Objeto de tipo Usuario.		
Precondiciones	Debe existir el objeto sessionFactory. Debe existir la base de datos y ser accesible. Debe existir un usuario con la clave primaria introducida.		
Pseudocódigo	Usuario usuario = session.uniqueResult(); return usuario;		
Secuencia de ejecución válida	Se invoca el método pasándole como argumento la clave primaria del usuario que se quiere buscar. Se comprueba que el objeto devuelto corresponde con el esperado.		

Tabla 184. PN-04-DAO.

PN-01-GESTION			
Objetivo	Guarda un objeto de tipo Recurso.		
Componente	Gestión	Clase	GestionIncidencia
Método	saveResource		
Entrada	Título, tipo, nombre original, puntos del área, fecha, descripción e identificador de la incidencia a la que pertenece.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto sessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	Recurso recurso = new Recurso(); recurso.titulo = titulo; recurso.nombreOriginal = nombreOriginal; recurso.puntos = puntos; recurso.fecha = fecha; recurso.descripcion = descripción; recurso.idPadre = idPadre; HibernateRecurso hr = new HibernateRecurso(); return hr.setRecurso(recurso);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 185. PN-01-GESTION.

PN-02-GESTION			
Objetivo	Guarda un objeto de tipo Subincidencia.		
Componente	Gestión	Clase	GestionIncidencia

Método	saveSubIncidencia
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción e identificador de la incidencia a la que pertenece.
Salida	Valor booleano que indica el resultado de la operación.
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.
Pseudocódigo	<pre> Subincidencia subincidencia = new Subincidencia(); subincidencia.titulo = titulo; subincidencia.tipo = tipo; subincidencia.subtipo = subtipo; subincidencia.fechaIni = fechaIni; subincidencia.fechaFin = fechaFin; subincidencia.puntos = puntos subincidencia.descripcion = descripción; subincidencia.idPadre = idPadre; HibernateSubIncidencia hsi = new HibernateSubIncidencia (); return hsi.setSubIncidencia(subincidencia); </pre>
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.

Tabla 186. PN-02-GESTION.

PN-03-GESTION			
Objetivo	Guarda un objeto de tipo Incidencia.		
Componente	Gestión	Clase	GestionIncidencia
Método	saveIncidencia		
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	<pre> Incidencia incidencia = new Incidencia (); incidencia.titulo = titulo; incidencia.tipo = tipo; incidencia.subtipo = subtipo; incidencia.fechaIni = fechaIni; incidencia.fechaFin = fechaFin; incidencia.puntos = puntos incidencia.descripcion = descripción; HibernateIncidencia hi = new HibernateIncidencia (); return hi.setIncidencia(incidencia); </pre>		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 187. PN-03-GESTION.

PN-04-GESTION

Objetivo	Elimina un objeto de tipo Recurso.		
Componente	Gestión	Clase	GestionIncidencia
Método	deleteResource		
Entrada	Identificador del reporte.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto sessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	HibernateRecurso hr = new HibernateRecurso(); return hr.deleteRecurso(id);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 188. PN-04-GESTION.

PN-05-GESTION			
Objetivo	Elimina un objeto de tipo Subincidencia.		
Componente	Gestión	Clase	GestionIncidencia
Método	deleteSubIncidencia		
Entrada	Identificador de la acción.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto sessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	HibernateSubIncidencia hsi = new HibernateSubIncidencia (); return hsi.deleteSubIncidencia(id);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 189. PN-05-GESTION.

PN-06-GESTION			
Objetivo	Elimina un objeto de tipo Incidencia.		
Componente	Gestión	Clase	GestionIncidencia
Método	deleteIncidencia		
Entrada	Identificador de la incidencia.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto sessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	HibernateIncidencia hi = new HibernateIncidencia (); return hi.deleteIncidencia(id);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 190. PN-06-GESTION.

PN-07-GESTION	
Objetivo	Actualiza un objeto de tipo Incidencia.

Componente	Gestión	Clase	GestionIncidencia
Método	updateIncidencia		
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción, identificador de la incidencia.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	Incidencia incidencia = new Incidencia (id); incidencia.titulo = titulo; incidencia.tipo = tipo; incidencia.subtipo = subtipo; incidencia.fechaIni = fechaIni; incidencia.fechaFin = fechaFin; incidencia.puntos = puntos incidencia.descripcion = descripción; HibernateIncidencia hi = new HibernateIncidencia (); return hi.updateIncidencia(incidencia);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 191. PN-07-GESTION.

PN-08-GESTION			
Objetivo	Actualiza un objeto de tipo Subincidencia.		
Componente	Gestión	Clase	GestionIncidencia
Método	updateSubIncidencia		
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción e identificador de la acción.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	Subincidencia subincidencia = new Subincidencia(id); subincidencia.titulo = titulo; subincidencia.tipo = tipo; subincidencia.subtipo = subtipo; subincidencia.fechaIni = fechaIni; subincidencia.fechaFin = fechaFin; subincidencia.puntos = puntos subincidencia.descripcion = descripción; HibernateSubIncidencia hsi = new HibernateSubIncidencia (); return hsi.updateSubIncidencia(subincidencia);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 192. PN-08-GESTION.

PN-09-GESTION

Objetivo	Actualiza un objeto de tipo Recurso.		
Componente	Gestión	Clase	GestionIncidencia
Método	updateResource		
Entrada	Título, tipo, nombre original, puntos del área, fecha, descripción e identificador del reporte.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	Recurso recurso = new Recurso(id); recurso.titulo = titulo; recurso.nombreOriginal = nombreOriginal; recurso.puntos = puntos; recurso.fecha = fecha recurso.descripcion = descripción; HibernateRecurso hr = new HibernateRecurso(); return hr.updateRecurso(recurso);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 193. PN-09-GESTION.

PN-10-GESTION			
Objetivo	Guarda un objeto de tipo Usuario.		
Componente	Gestión	Clase	GestionUsuarios
Método	saveUsuario		
Entrada	Nick, perfil, contraseña, nombre, apellidos, teléfono y e-mail.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	Usuario usuario = new Usuario(); usuario.nick = nick; usuario.contraseña = contraseña; usuario.nombre = nombre; usuario.apellidos = apellidos usuario.telefono = telefono; usuario.email = email; HibernateUsuario hu = new HibernateUsuario(); return hu.setUsuario(usuario);		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 194. PN-10-GESTION.

PN-11-GESTION			
Objetivo	Actualiza un objeto de tipo Usuario.		
Componente	Gestión	Clase	GestionUsuarios

Método	updateUsuario
Entrada	Nick, perfil, contraseña, nombre, apellidos, teléfono y e-mail.
Salida	Valor booleano que indica el resultado de la operación.
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.
Pseudocódigo	<pre> Usuario usuario = new Usuario(); usuario.nick = nick; usuario.contraseña = contraseña; usuario.nombre = nombre; usuario.apellidos = apellidos; usuario.telefono = telefono; usuario.email = email; HibernateUsuario hu = new HibernateUsuario(); return hu.updateUsuario(usuario); </pre>
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.

Tabla 195. PN-11-GESTION.

PN-12-GESTION			
Objetivo	Elimina un objeto de tipo Usuario.		
Componente	Gestión	Clase	GestionUsuarios
Método	deleteUsuario		
Entrada	Nick del usuario.		
Salida	Valor booleano que indica el resultado de la operación.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	<pre> HibernateUsuario hu = new HibernateUsuario(); return hu.deleteUsuario(nick); </pre>		
Secuencia de ejecución válida	Se invoca el método pasándole todos los argumentos. Se comprueba que el objeto se ha guardado en la base de datos con los valores pasados.		

Tabla 196. PN-12-GESTION.

PN-01-KML			
Objetivo	Genera los documentos KML para los timemaps de la administración de las incidencias.		
Componente	Gestión	Clase	AdminEventsKML
Método	generar		
Entrada	void		
Salida	Lista de documentos KML generados.		
Precondiciones	Debe existir el objeto SessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	<pre> List<Incidencia> lista = getIncidencias(); List<ElementoNombreKML> listaDoc; For lista:elemento </pre>		

	<pre>listaDoc.add(elemento.getNombre()); listaDoc.add(nombreDocumento); return listaDoc;</pre>
Secuencia de ejecución válida	<p>Se invoca el método.</p> <p>Se comprueba que se han generado los documentos KML tal y como indica el objeto devuelto.</p>

Tabla 197. PN-01-KML.

PN-02-KML			
Objetivo	Genera los documentos KML para los timemaps de la comparación de datos.		
Componente	Gestión	Clase	ComparisonKML
Método	generar		
Entrada	void		
Salida	Lista de documentos KML generados.		
Precondiciones	<p>Debe existir el objeto SessionFactory.</p> <p>Debe existir la base de datos y ser accesible.</p>		
Pseudocódigo	<pre>List<Incidencia> lista = getIncidencias(); List<ElementoNombreKML> listaDoc; For lista:elemento listaDoc.add(elemento.getNombre()); listaDoc.add(nombreDocumento); return listaDoc;</pre>		
Secuencia de ejecución válida	<p>Se invoca el método.</p> <p>Se comprueba que se han generado los documentos KML tal y como indica el objeto devuelto.</p>		

Tabla 198. PN-02-KML.

PN-03-KML			
Objetivo	Genera los documentos KML para los timemaps de la vista en detalle de una incidencia.		
Componente	Gestión	Clase	DetailKML
Método	generar		
Entrada	void		
Salida	Lista de documentos KML generados.		
Precondiciones	<p>Debe existir el objeto SessionFactory.</p> <p>Debe existir la base de datos y ser accesible.</p>		
Pseudocódigo	<pre>List<Incidencia> lista = getIncidencias(); List<ElementoNombreKML> listaDoc; For lista:elemento listaDoc.add(elemento.getNombre()); listaDoc.add(nombreDocumento); return listaDoc;</pre>		
Secuencia de ejecución válida	<p>Se invoca el método.</p> <p>Se comprueba que se han generado los documentos KML tal y como indica el objeto devuelto.</p>		

Tabla 199. PN-03-KML.

PN-04-KML			
Objetivo	Genera los documentos KML para los timemaps de la vista general de la aplicación.		
Componente	Gestión	Clase	GeneralKML
Método	generar		
Entrada	void		
Salida	Lista de documentos KML generados.		
Precondiciones	Debe existir el objeto sessionFactory. Debe existir la base de datos y ser accesible.		
Pseudocódigo	<pre>List<Incidencia> lista = getIncidencias(); List<ElementoNombreKML> listaDoc; For lista:elemento listaDoc.add(elemento.getNombre()); listaDoc.add(nombreDocumento); return listaDoc;</pre>		
Secuencia de ejecución válida	Se invoca el método. Se comprueba que se han generado los documentos KML tal y como indica el objeto devuelto.		

Tabla 200. PN-04-KML.

Pruebas de Integración

Pruebas de integración: forman la fase del testeo de software en la cual módulos individuales de software son combinados y testeados como un grupo, verificando que todas las partes del software funcionan unidas.

PI-01	
Descripción	Dar de alta un usuario.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	SubmitUserBean, GestionUsuarios, HibernateUsuarios, HibernatePerfil, HibernateUtil, UserBean, Perfil, Usuarios.
Entrada	Nick, perfil, contraseña, nombre, apellidos, teléfono y e-mail.
Salida	Redirección a la página de administración de usuarios. Aparecerá el nuevo usuario en la lista de usuarios.
Precondiciones	No puede existir un usuario con el mismo Nick.
Post-condiciones	N/A.

Tabla 201. PI-01.

PI-02	
Descripción	Dar de baja un usuario.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	SubmitUserBean, GestionUsuarios, HibernateUsuarios, HibernatePerfil, HibernateUtil, UserBean, Perfil, Usuarios.
Entrada	Nick del usuario a borrar.
Salida	Redirección a la página de administración de usuarios. No aparecerá el usuario borrado en la lista de usuarios.
Precondiciones	El Nick de usuario debe existir.

Post-condiciones	Se eliminarán todos los datos del usuario.
-------------------------	--

Tabla 202. PI-02.

PI-03	
Descripción	Acceso al sistema.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionUsuarios, HibernateUsuarios, HibernatePerfil, HibernateUtil, UserBean, Perfil, Usuarios.
Entrada	Nick y contraseña del usuario.
Salida	Ventana principal del sistema.
Precondiciones	El usuario debe estar registrado en el sistema.
Post-condiciones	N/A.

Tabla 203. PI-03.

PI-04	
Descripción	Cerrar sesión.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionUsuarios, HibernateUsuarios, HibernatePerfil, HibernateUtil, UserBean, Perfil, Usuarios.
Entrada	Nick del usuario que quiere cerrar la sesión.
Salida	Ventana de login del sistema.
Precondiciones	El usuario debe estar registrado en el sistema. El usuario debe haber accedido al sistema.
Post-condiciones	N/A.

Tabla 204. PI-04.

PI-05	
Descripción	Modificar datos usuario.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionUsuarios, HibernateUsuarios, HibernatePerfil, HibernateUtil, UserBean, Perfil, Usuarios.
Entrada	Nick, perfil, contraseña, nombre, apellidos, teléfono y e-mail.
Salida	Redirección a la página de administración de usuarios. Aparecerá el usuario con los datos modificados en la lista de usuarios.
Precondiciones	El usuario debe estar registrado en el sistema. No se puede cambiar el Nick si este ya está siendo utilizado por otro usuario.
Post-condiciones	Los datos del usuario han cambiado.

Tabla 205. PI-05.

PI-06	
Descripción	Crear incidencia.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, SubmitEventBean, HibernateIncidencia, HibernateTipoSubTipo, HibernateUtil, Incidencia, Tipo.
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, descripción y

	ubicación.
Salida	Redirección a la página principal del sistema dónde aparecerá la nueva incidencia añadida.
Precondiciones	El usuario debe haber accedido al sistema.
Post-condiciones	Añadida la nueva incidencia a la base de datos, pudiendo verla en la pantalla general del sistema.

Tabla 206. PI-06.

PI-07	
Descripción	Visualizar incidencia.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoSubTipo, HibernateUtil, Incidencia, Tipo, DetailKML.
Entrada	Identificador de la incidencia a visualizar.
Salida	Redirección a la página en la que se verán los detalles de la incidencia, sus datos, sus acciones y reportes.
Precondiciones	El identificador de la incidencia debe existir. El usuario debe haber accedido al sistema.
Post-condiciones	N/A.

Tabla 207. PI-07.

PI-08	
Descripción	Modificar incidencia.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoSubTipo, HibernateUtil, Incidencia, Tipo, EditEventBean, AdminEventsKML.
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, descripción y ubicación.
Salida	Redirección a la página en la que se verán los detalles de la incidencia modificada con sus datos nuevos.
Precondiciones	La incidencia debe existir en la base de datos. El usuario debe haber accedido al sistema.
Post-condiciones	Los datos de la incidencia son modificados en la base de datos.

Tabla 208. PI-08.

PI-09	
Descripción	Borrar incidencia.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoSubTipo, HibernateRecurso, HibernateSubIncidencia, HibernateUtil, Incidencia, Subincidencia, Recurso, AdminEventsKML.
Entrada	Identificador de la incidencia a eliminar.
Salida	Redirección a la página principal del sistema.
Precondiciones	La incidencia debe existir en la base de datos. El usuario debe haber accedido al sistema.
Post-condiciones	Los datos de la incidencia son eliminados de la base de datos.

Tabla 209. PI-09.

PI-10	
Descripción	Crear acción.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, SubmitSubEventBean, HibernateIncidencia, HibernateTipoSubTipo, HibernateSubIncidencia, HibernateUtil, Incidencia, Tipo, Subincidencia.
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción e identificador de la acción.
Salida	Redirección a la página de detalles de la incidencia padre.
Precondiciones	La incidencia padre debe existir en la base de datos. El usuario debe haber accedido al sistema.
Post-condiciones	Creación de la acción en la base de datos.

Tabla 210. PI-10.

PI-11	
Descripción	Visualizar acción.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, SubmitSubEventBean, HibernateIncidencia, HibernateTipoSubTipo, HibernateSubIncidencia, HibernateUtil, Incidencia, Tipo, Subincidencia, DetailKML.
Entrada	Identificador de la acción a visualizar.
Salida	Visualización de la acción en la pantalla del navegador.
Precondiciones	El usuario debe haber accedido al sistema. La acción debe existir en la base de datos.
Post-condiciones	N/A.

Tabla 211. PI-11.

PI-12	
Descripción	Modificar incidencia.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, EditSubEventBean, HibernateIncidencia, HibernateTipoSubTipo, HibernateSubIncidencia, HibernateUtil, Incidencia, Tipo, Subincidencia, DetailKML, AdminEventsKML.
Entrada	Título, tipo, subtipo, fecha de inicio, fecha de fin, puntos del área, descripción e identificador de la acción.
Salida	Visualización de la acción en la pantalla del navegador con los nuevos datos.
Precondiciones	El usuario debe haber accedido al sistema. La acción debe existir en la base de datos.
Post-condiciones	Los datos de la base de datos son modificados.

Tabla 212. PI-12.

PI-13	
Descripción	Borrar incidencia.

Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, EditSubEventBean, HibernateIncidencia, HibernateTipoSubTipo, HibernateSubIncidencia, HibernateUtil, Incidencia, Tipo, Subincidencia, DetailKML, AdminEventsKML.
Entrada	Identificador de la acción a borrar.
Salida	Redirección a los detalles de la incidencia padre.
Precondiciones	El usuario debe haber accedido al sistema. La acción debe existir en la base de datos.
Post-condiciones	La acción es eliminada de la base de datos.

Tabla 213. PI-13.

PI-14	
Descripción	Crear reporte.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, SubmitResourceBean, HibernateIncidencia, HibernateTipoRecurso, HibernateRecurso, HibernateUtil, Incidencia, TipoRecurso, Recurso.
Entrada	Título, tipo, nombre original, puntos del área, fecha, descripción e identificador de la incidencia a la que pertenece.
Salida	Redirección a los detalles de la incidencia padre.
Precondiciones	El usuario debe haber accedido al sistema.
Post-condiciones	El reporte es añadido a la base de datos.

Tabla 214. PI-14.

PI-15	
Descripción	Visualizar reporte.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoRecurso, HibernateRecurso, HibernateUtil, Incidencia, TipoRecurso, Recurso.
Entrada	Identificador del recurso.
Salida	Datos del recurso mostrados por pantalla. El recurso puede ser descargado.
Precondiciones	El usuario debe haber accedido al sistema. El reporte debe existir en la base de datos.
Post-condiciones	N/A.

Tabla 215. PI-15.

PI-16	
Descripción	Modificar reporte.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, EditResourceBean, HibernateIncidencia, HibernateTipoRecurso, HibernateRecurso, HibernateUtil, Incidencia, TipoRecurso, Recurso.
Entrada	Título, tipo, nombre original, puntos del área, fecha, descripción e identificador de la incidencia a la que pertenece.
Salida	Datos del recurso mostrados por pantalla. El recurso puede ser

	descargado.
Precondiciones	El usuario debe haber accedido al sistema. El reporte debe existir en la base de datos.
Post-condiciones	Los datos del reporte son modificados en la base de datos.

Tabla 216. PI-16.

PI-17	
Descripción	Borrar reporte.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, EditResourceBean, HibernateIncidencia, HibernateRecurso, HibernateUtil, Incidencia, Recurso.
Entrada	Identificador del reporte a borrar.
Salida	Redirección a los detalles de la incidencia padre.
Precondiciones	El usuario debe haber accedido al sistema. El reporte debe existir en la base de datos.
Post-condiciones	El reporte es eliminado de la base de datos.

Tabla 217. PI-17.

PI-18	
Descripción	Visualizar todas las incidencias mediante un mapa.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoSubTipo, HibernateUtil, Incidencia,Tipo, Subtipo, GeneralKML.
Entrada	Identificador de todas las incidencias almacenadas en la base de datos.
Salida	El sistema muestra un mapa en la que aparecen todas las incidencias almacenadas.
Precondiciones	El usuario debe haber accedido al sistema. Las incidencias deben existir en la base de datos.
Post-condiciones	N/A.

Tabla 218. PI-18.

PI-19	
Descripción	Visualizar todas las incidencias mediante una lista.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoSubTipo, HibernateUtil, Incidencia,Tipo, Subtipo, GeneralKML.
Entrada	Identificador de todas las incidencias almacenadas en la base de datos.
Salida	El sistema muestra una lista en la que aparecen todas las incidencias almacenadas.
Precondiciones	El usuario debe haber accedido al sistema. Las incidencias deben existir en la base de datos.
Post-condiciones	N/A.

Tabla 219. PI-19.

PI-20	
Descripción	Comparación de colección de incidencias.
Subsistemas	Vista, Controlador, Modelo, Infraestructura.
Clases	GestionIncidencia, HibernateIncidencia, HibernateTipoSubTipo, HibernateUtil, Incidencia,Tipo, Subtipo, ComparisonKML.
Entrada	Identificador de todas las incidencias almacenadas en la base de datos.
Salida	El sistema muestra dos mapas con los mismos datos. Se habilitan filtros por tipo y subtipo para poder realizar comparaciones.
Precondiciones	El usuario debe haber accedido al sistema. Las incidencias deben existir en la base de datos.
Post-condiciones	N/A.

Tabla 220. PI-20.

4.3 El producto del desarrollo

Se ha generado una aplicación web que tiene como principales funciones las siguientes:

- Dotar de información espacio/temporal visual a un conjunto de incidencias.
- Separar la información objetiva de la información subjetiva asociada a cada incidencia.
- Posibilidad de comparar dos conjuntos de datos mediante el uso de mapas y líneas temporales.

A continuación se explicará los pasos que hay que seguir para instalar el sistema.

1. Comprobar que se encuentra instalada una versión de MySQL igual o superior a la versión 5.5 (aplicación optimizada para la versión 5.5).
2. Ejecutar sobre la base de datos los scripts SQL de creación de base de datos, creación de usuario y actualización de valores de tablas en el siguiente orden:
 - a. `'tablas.sql'`.
 - b. `'create_user.sql'`.
 - c. `'data.sql'`.
3. Comprobar que se encuentra instalada una versión de Java igual o superior a la versión 1.6.
4. Generar y desplegar el archivo WAR de la aplicación web en un servidor Apache Tomcat (aplicación optimizada para la versión 7.0).
5. Acceder desde un navegador web a la dirección web donde se haya desplegado la aplicación web.

Si el sistema se ha instalado correctamente, debería aparecer la siguiente pantalla en el navegador web:



Figura 28. Pantalla de inicio de la aplicación.

Para acceder al sistema sólo hay que introducir un usuario y contraseña válidos. Por defecto, el usuario disponible es:

- User: sigi
- Password: sigi

Una vez introducidos los datos se podrá utilizar la herramienta de forma normal. La pantalla que debe aparecer una vez introducido los datos anteriores debería ser la siguiente (*para saber cómo utilizar la aplicación se especifica el manual de usuario en el apartado Anexo III. Manual de usuario*):

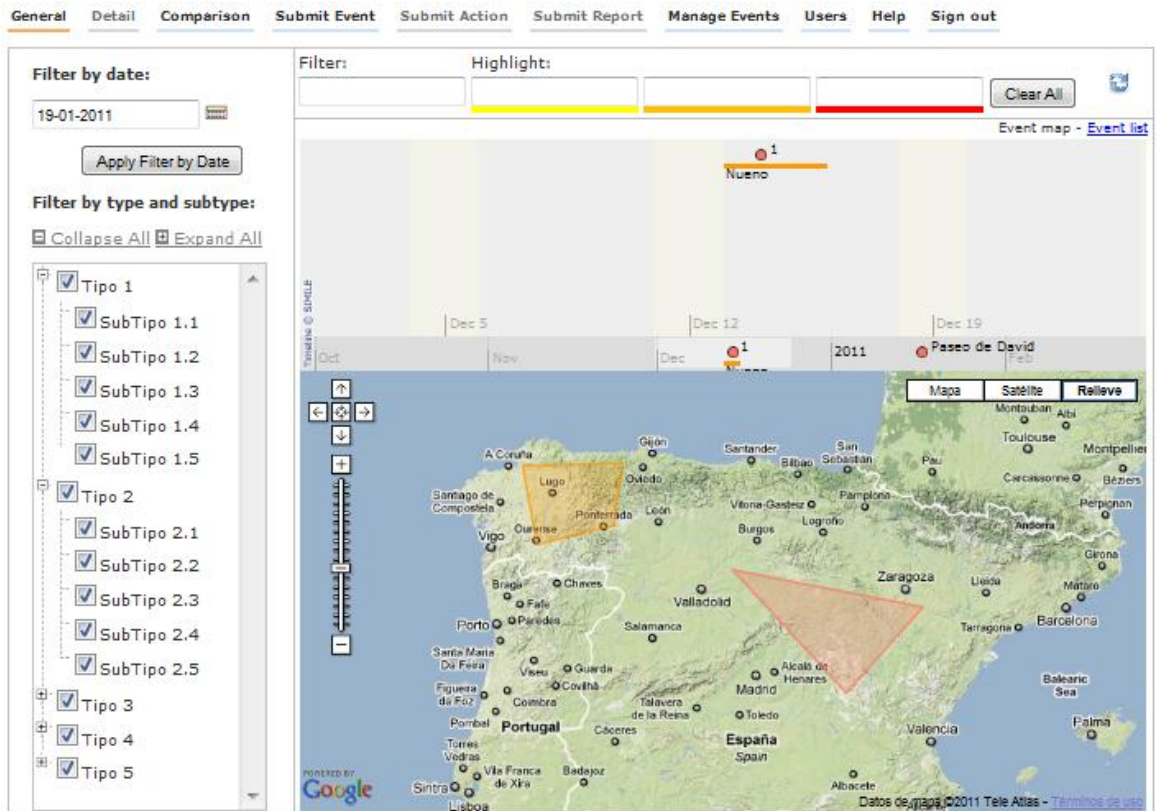


Figura 29. Pantalla General de la aplicación.

5 Evaluación

En este apartado se indicará cómo se va a evaluar la aplicación para indicar si el sistema construido contiene los objetivos buscados. Seguidamente se realizará un análisis de los resultados obtenidos y se alcanzará una conclusión que demuestre la validez del sistema.

5.1 Proceso de evaluación

Plan de pruebas

La aplicación desarrollada se evaluará con un conjunto de pruebas que validarán todos los requisitos funcionales establecidos. Las principales características que la aplicación debe cumplir son las siguientes y deben estar presentes durante todo el proceso de validación:

- Dotar de información espacio/temporal visual a un conjunto de incidencias.
- Diferenciar los distintos tipos de reportes asociados a cada una de las incidencias.
- Posibilidad de comparar de forma visual dos conjuntos de datos utilizando su información espacio/temporal.

Es igual de importante validar los tres objetivos anteriores como garantizar que todos los requisitos funcionales han sido implementados y para ello se han elaborado las pruebas que se muestran en el siguiente apartado.

Casos de prueba

A continuación se muestran todas las pruebas realizadas. Cada prueba dispone de un identificador, un objetivo, el procedimiento que se debe seguir y el resultado obtenido.

P-01	
Objetivo	Comprobar que se muestran todas las incidencias mediante un mapa y línea temporal en la vista general.
Procedimiento	<ul style="list-style-type: none">• Pulsar el botón “General” del menú superior.• Acceder a la vista de mapa de incidencias.• Desplazarse por la línea temporal y comprobar que existen todas las incidencias almacenadas.

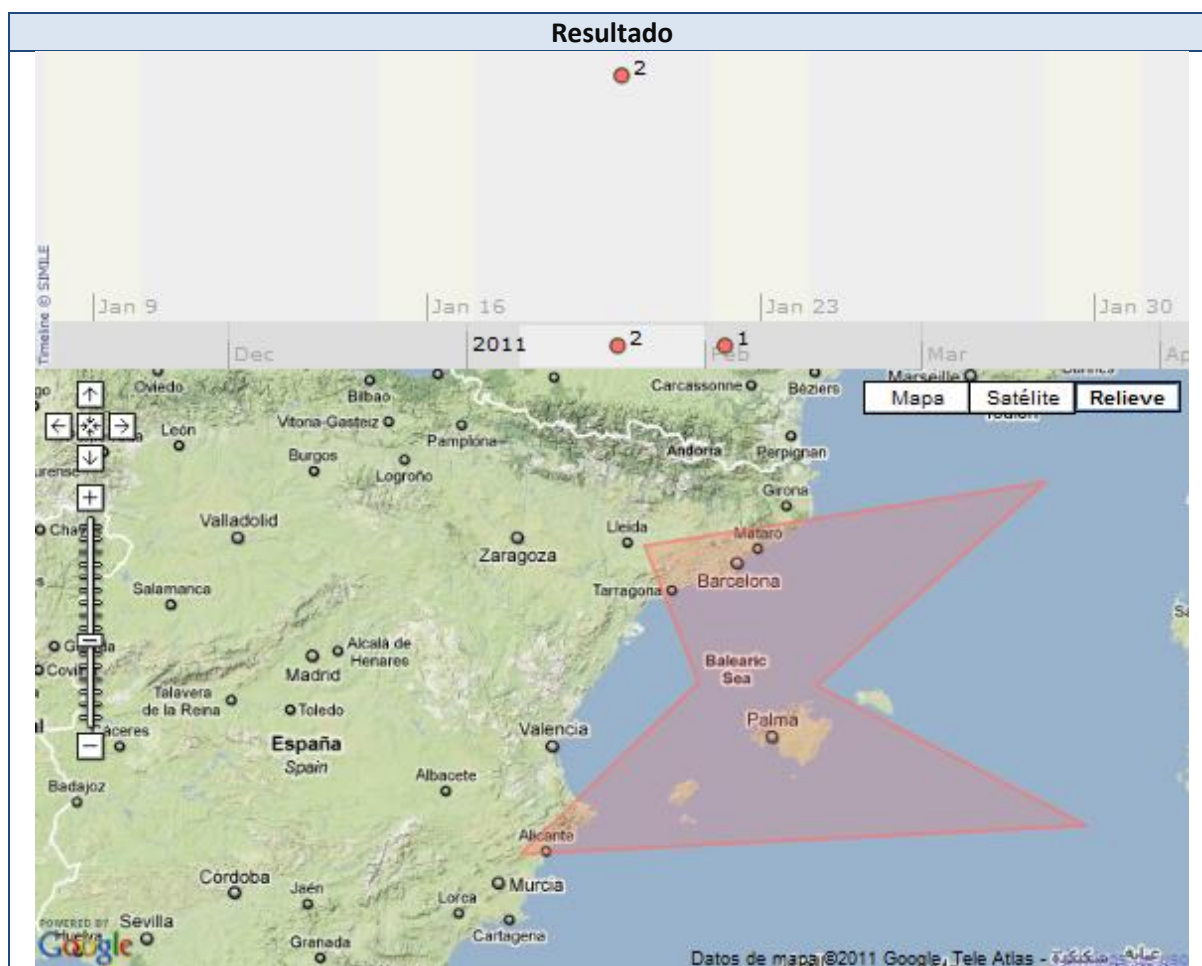


Tabla 221. Prueba P-01.

P-02					
Objetivo	Comprobar que se muestran todas las incidencias mediante una lista en la vista general.				
Procedimiento	<ul style="list-style-type: none">• Pulsar el botón “General” del menú superior.• Acceder a la vista de lista de incidencias.• Comprobar que existen todas las incidencias en la lista mostrada.				
Resultado					
Title	Type	Subtype	From	To	Actions
1bis	Tipo 1	SubTipo 1.1	03-02-2011	In progress	Detail
2	Tipo 1	SubTipo 1.1	20-01-2011	In progress	Detail
111111	Tipo 1	SubTipo 1.1	10-02-2011	In progress	Detail

Tabla 222. Prueba P-02.

P-03	
Objetivo	Comprobar que el filtro por fecha desplaza la línea temporal al lugar solicitado en la vista general.
Procedimiento	<ul style="list-style-type: none"> Pulsar el botón “General” del menú superior. Acceder a la vista de mapa de incidencias.

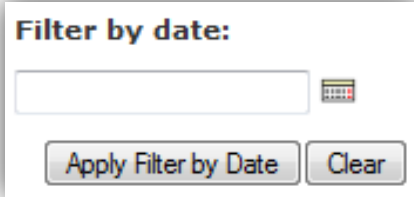
	<ul style="list-style-type: none"> • Introducir la fecha 03/02/2011 y pulsar el botón “<i>Apply Filter by Date</i>”. • Comprobar que la línea temporal se ha desplazado a la fecha introducida.
Resultado	
	

Tabla 223. Prueba P-03.

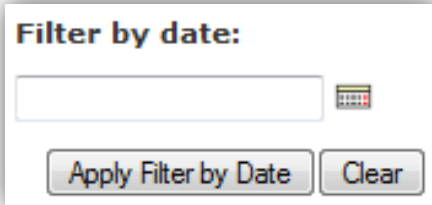
P-04	
Objetivo	Comprobar que el filtro por fecha resalta las incidencias que se encuentra en la lista de la vista general.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “<i>General</i>” del menú superior. • Acceder a la vista de lista de incidencias. • Introducir la fecha 03/02/2011 y pulsar el botón “<i>Apply Filter by Date</i>”. • Comprobar que las incidencias que tienen la fecha de inicio introducida son resaltadas.
Resultado	
	

Tabla 224. Prueba P-04.

P-05	
Objetivo	Comprobar que en el mapa de la vista general al seleccionar un tipo o subtipo se muestran las incidencias que tenga dicho tipo o subtipo. Comprobar que al deseleccionar un tipo o subtipo se ocultan las incidencias que tengan dicho tipo o subtipo.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “<i>General</i>” del menú superior. • Acceder a la vista de mapa de incidencias. • Marcar el primer tipo y desmarcar el segundo subtipo. • Comprobar que las incidencias del primer tipo aparecen en el mapa y la línea temporal y que las incidencias que tengan como subtipo el subtipo desmarcado son ocultados en la línea temporal y mapa.

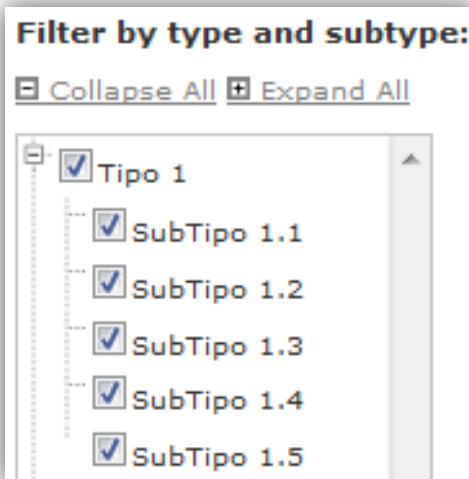
Resultado


Tabla 225. Prueba P-05.

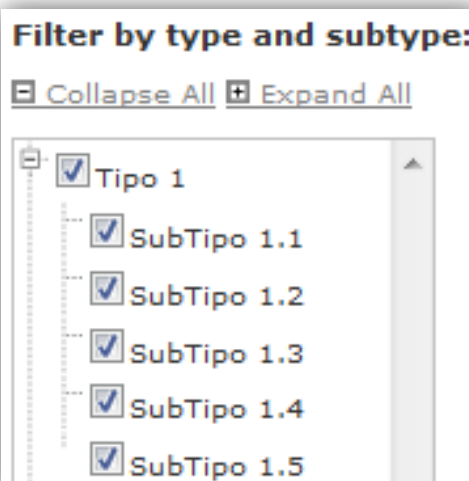
P-06	
Objetivo	Comprobar que en la lista de la vista general al seleccionar un tipo o subtipo se muestran las incidencias que tenga dicho tipo o subtipo. Comprobar que al deseleccionar un tipo o subtipo se ocultan las incidencias que tengan dicho tipo o subtipo.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “General” del menú superior. • Acceder a la vista de lista de incidencias. • Marcar el primer tipo y desmarcar el segundo subtipo. • Comprobar que las incidencias del primer tipo aparecen en la lista y las incidencias del segundo subtipo no aparecen en la lista.
Resultado	
	

Tabla 226. Prueba P-06.


P-07	
Objetivo	Comprobar que en el mapa de la vista general al introducir el título de una incidencia en el filtro por texto ésta es resaltada.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “General” del menú superior. • Acceder a la vista de mapa de incidencias. • Introducir el título de una incidencia en el filtro por texto. • Comprobar que el título introducido resalta la incidencia que tiene dicho título.
Resultado	
	

Tabla 227. Prueba P-07.

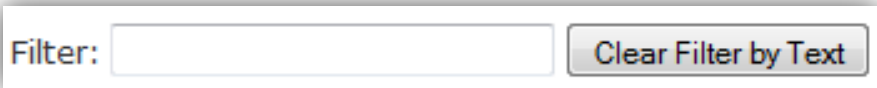
P-08	
Objetivo	Comprobar que en la lista de la vista general al introducir el título de una incidencia en el filtro por texto ésta es resaltada.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “General” del menú superior. • Acceder a la vista de lista de incidencias. • Introducir el título de una incidencia en el filtro por texto. • Comprobar que el título introducido resalta la incidencia que tiene dicho título.
Resultado	
	

Tabla 228. Prueba P-08.

P-09	
Objetivo	Comprobar que se pueden comparar dos colecciones de datos mediante dos mapas y líneas temporales.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “Comparison” del menú superior. • Comprobar que aparecen dos mapas y dos líneas temporales. • Comprobar que ambos mapas son independientes. • Comprobar que ambas líneas temporales son independientes.

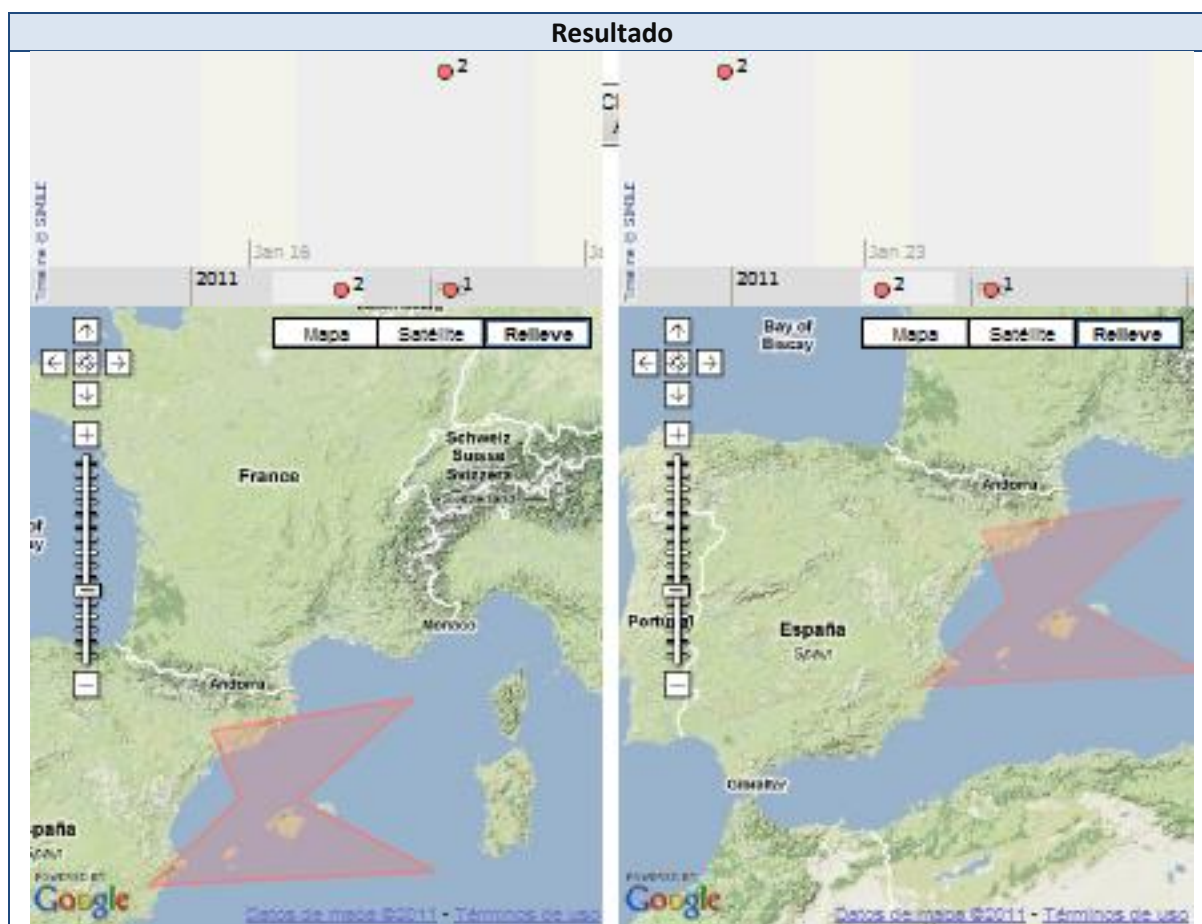


Tabla 229. Prueba P-09.

P-10	
Objetivo	Comprobar que el filtro por fecha desplaza las líneas temporales de la vista de comparación.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “Comparison” del menú superior. • Introducir la fecha 03/02/2011 y pulsar el botón “Apply TimeMap 1”. • Introducir la fecha 04/02/2011 y pulsar el botón “Apply TimeMap 2”. • Comprobar que las líneas temporales se han desplazado a la fecha introducida.
Resultado	

Tabla 230. Prueba P-10.

P-11	
Objetivo	Comprobar que el filtro por tipo y subtipo oculta o muestra las distintas incidencias de la vista comparación.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “Comparison” del menú superior.

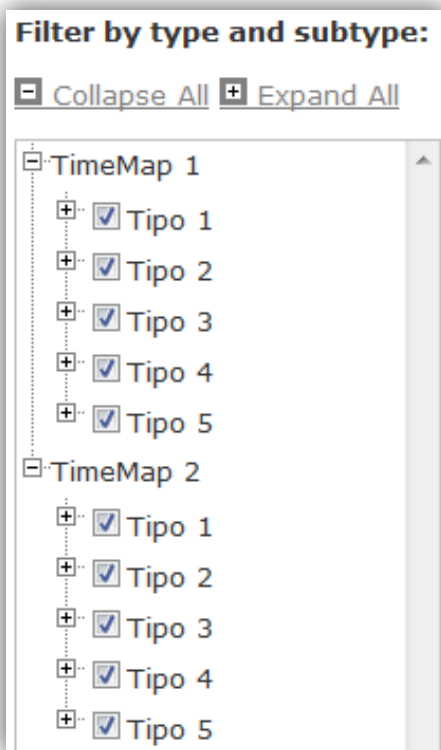
	<ul style="list-style-type: none"> Desmarcar el primer tipo del primer mapa. Desmarcar el segundo tipo del segundo mapa. Comprobar que las incidencias del primer mapa que sean del primer tipo no aparecen en el primer timemap. Comprobar que las incidencias del segundo timemap que sean del segundo tipo no aparecen en el primer timemap.
Resultado	
	

Tabla 231. Prueba P-11.

P-12	
Objetivo	Comprobar que el filtro por texto de la vista de comparación resalta las incidencias que contengan en su título el texto introducido.
Procedimiento	<ul style="list-style-type: none"> Pulsar el botón “<i>Comparison</i>” del menú superior. Introducir una palabra que se sepa que existe en el título de una incidencia. Comprobar que las incidencias que contengan el texto introducido son resaltadas.
Resultado	
Filter TimeMap 1: <input type="text"/> Highlight TimeMap 1: <input type="text"/> <input type="button" value="Clear All"/>	Filter TimeMap 2: <input type="text"/> Highlight TimeMap 2: <input type="text"/> <input type="button" value="Clear All"/>

Tabla 232. Prueba P-12.

P-13

Objetivo	Comprobar que se pueden guardar incidencias en el sistema.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"Submit Event"</i> del menú superior. • Introducir un título, tipo, subtipo y fecha inicial. La fecha final y descripción son opcionales. • Introducir una ubicación y pulsar el botón <i>"Close Polyshape"</i>. • Pulsar el botón <i>"Save Event"</i>. • Pulsar el botón <i>"General"</i> del menú superior. • Comprobar que la incidencia guardada aparece en el mapa o lista.
Resultado	La incidencia se agregó correctamente al sistema.

Tabla 233. Prueba P-13.

P-14	
Objetivo	Comprobar que se pueden guardar acciones en el sistema.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"General"</i> del menú superior. • Seleccionar una incidencia y pulsar sobre su botón <i>"Detail"</i>. • El sistema cargará la vista de detalle de la incidencia. • Pulsar el botón <i>"Submit Action"</i> del menú superior que ha quedado ahora activo. • Introducir un título, tipo, subtipo y fecha inicial. La fecha final y descripción son opcionales. • Introducir una ubicación y pulsar el botón <i>"Close Polyshape"</i>. • Pulsar el botón <i>"Save Event"</i>. El sistema cargará la vista en detalle de la incidencia. • Comprobar que la nueva acción introducida aparece en la lista.
Resultado	La acción se agregó correctamente al sistema.

Tabla 234. Prueba P-14.

P-15	
Objetivo	Comprobar que se pueden guardar reportes en el sistema.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"General"</i> del menú superior. • Seleccionar una incidencia y pulsar sobre su botón <i>"Detail"</i>. • El sistema cargará la vista de detalle de la incidencia. • Pulsar el botón <i>"Submit Report"</i> del menú superior que ha quedado ahora activo. • Introducir un título, tipo, fecha y seleccionar un archivo. La descripción es opcional. • Introducir una ubicación y pulsar el botón <i>"Save Marker"</i>. • Pulsar el botón <i>"Save Resource"</i>. • El sistema mostrará un mensaje con el resultado de la operación. • Pulsar el botón <i>"Detail"</i> del menú superior. • Pulsar el botón <i>"Dossier"</i> del menú lateral izquierdo. • Comprobar que el nuevo reporte introducido aparece en el visor de reportes.
Resultado	El reporte se agregó correctamente al sistema.

Tabla 235. Prueba P15.

P-16	
Objetivo	Comprobar que se pueden modificar los datos de una incidencia.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “<i>Manage Events</i>” del menú superior. • Elegir una incidencia y pulsar sobre su botón “<i>Edit Event</i>”. • El sistema cargará la vista de edición de la incidencia. • Introducir un nuevo título, tipo, subtipo, fecha inicial, fecha final, descripción o ubicación. • En el caso que se quiera actualizar la ubicación se debe marcar el check “<i>Save new location</i>” situado en el inferior de la ventana. • Pulsar el botón “<i>Save Event</i>”. • El sistema mostrará la pantalla de administración de eventos con los cambios realizados. • Comprobar que la incidencia tiene los cambios realizados.
Resultado	La incidencia se modificó correctamente.

Tabla 236. Prueba P-16.

P-17	
Objetivo	Comprobar que se pueden modificar los datos de una acción.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “<i>Manage Events</i>” del menú superior. • Elegir una acción y pulsar sobre su botón “<i>Edit Action</i>”. • El sistema cargará la vista de edición de la acción. • Introducir un nuevo título, tipo, subtipo, fecha inicial, fecha final, descripción o ubicación. • En el caso que se quiera actualizar la ubicación se debe marcar el check “<i>Save new location</i>” situado en el inferior de la ventana. • Pulsar el botón “<i>Save Action</i>”. • El sistema mostrará la pantalla de administración de eventos con los cambios realizados. • Comprobar que la acción tiene los cambios realizados.
Resultado	La acción se modificó correctamente.

Tabla 237. Prueba P-17.

P-18	
Objetivo	Comprobar que se pueden modificar los datos de un reporte.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón “<i>Manage Events</i>” del menú superior. • Elegir una acción y pulsar sobre su botón “<i>Edit Report</i>”. • El sistema cargará la vista de edición del reporte. • Introducir un nuevo título, tipo, fecha, descripción, ubicación o archivo. • En el caso que se quiera actualizar la ubicación se debe marcar el check “<i>Save new location</i>” situado en el inferior de la ventana. • En el caso que se quiera actualizar el archivo se debe marcar el check “<i>Save new file</i>” situado en el inferior de la ventana. • Pulsar el botón “<i>Save Report</i>”. • El sistema mostrará la pantalla de administración de eventos con los cambios realizados. • Comprobar que el reporte tiene los cambios realizados.

Resultado	El reporte se modificó correctamente.
------------------	---------------------------------------

Tabla 238. Prueba P-18.

P-19	
Objetivo	Comprobar que se puede eliminar una incidencia.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"Manage Events"</i> del menú superior. • Elegir una incidencia y pulsar sobre su botón <i>"Delete Event"</i>. • El sistema mostrará la pantalla de administración de eventos con los cambios realizados. • Comprobar que la incidencia eliminada ya no aparece en la lista. • Comprobar que no existe ninguna acción o reporte relacionados con la incidencia eliminada.
Resultado	La incidencia se eliminó correctamente.

Tabla 239. Prueba P-19.

P-20	
Objetivo	Comprobar que se puede eliminar una acción.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"Manage Events"</i> del menú superior. • Elegir una incidencia y pulsar sobre su botón <i>"Delete Action"</i>. • El sistema mostrará la pantalla de administración de eventos con los cambios realizados. • Comprobar que la acción eliminada ya no aparece en la lista.
Resultado	La acción se eliminó correctamente.

Tabla 240. Prueba P-20.

P-21	
Objetivo	Comprobar que se puede eliminar un reporte.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"Manage Events"</i> del menú superior. • Elegir una incidencia y pulsar sobre su botón <i>"Delete Report"</i>. • El sistema mostrará la pantalla de administración de eventos con los cambios realizados. • Comprobar que el reporte eliminado ya no aparece en la lista.
Resultado	El reporte se eliminó correctamente.

Tabla 241. Prueba P-21.

P-22	
Objetivo	Comprobar que se puede crear un nuevo usuario.
Procedimiento	<ul style="list-style-type: none"> • Pulsar el botón <i>"Users"</i> del menú superior. • Pulsar sobre su botón <i>"Submit User"</i>. • Introducir un nombre de usuario, contraseña y perfil. • Introducir un nombre, apellidos, número de teléfono y e-mail. • Pulsar <i>"Save User"</i>. • Si el nombre de usuario ya existiese el sistema indicará que se debe cambiar. • En caso de que se pueda crear el usuario el sistema cargará la pantalla de usuarios.

	<ul style="list-style-type: none"> Comprobar que el nuevo usuario aparece en la lista.
Resultado	El usuario se creó correctamente.

Tabla 242. Prueba P-22.

P-23	
Objetivo	Comprobar que se puede modificar los datos de un usuario.
Procedimiento	<ul style="list-style-type: none"> Pulsar el botón “Users” del menú superior. Elegir un usuario y pulsar sobre su botón “Edit User”. Introducir un nuevo nombre de usuario, contraseña y perfil. Introducir un nuevo nombre, apellidos, número de teléfono y e-mail. Pulsar “Save User”. Si el nombre de usuario ya existiese el sistema indicará que se debe cambiar. En caso de que se puedan modificar los datos del usuario el sistema cargará la pantalla de usuarios. Comprobar que los datos del usuario han cambiado.
Resultado	El usuario se modificó correctamente.

Tabla 243. Prueba P-23.

P-24	
Objetivo	Comprobar que se puede eliminar un usuario.
Procedimiento	<ul style="list-style-type: none"> Pulsar el botón “Users” del menú superior. Elegir un usuario y pulsar sobre su botón “Delete User”. El sistema cargará la lista de usuarios. Comprobar que el usuario ha sido eliminado y no aparece en la lista.
Resultado	El usuario se eliminó correctamente.

Tabla 244. Prueba P-24.

P-25	
Objetivo	Comprobar que se pueden consultar las acciones de una incidencia.
Procedimiento	<ul style="list-style-type: none"> Pulsar el botón “General” del menú superior. Elegir una incidencia y pulsar sobre su botón “Detail”. El sistema cargará la lista de acciones. Comprobar que aparecen todas las acciones de la incidencia en la lista, en el mapa y en la línea temporal.
Resultado	Las acciones se consultaron correctamente.

Tabla 245. Prueba P-25.

P-26	
Objetivo	Comprobar que se pueden consultar los reportes de una incidencia.
Procedimiento	<ul style="list-style-type: none"> Pulsar el botón “General” del menú superior. Elegir una incidencia y pulsar sobre su botón “Detail”. Pulsar el botón “Dossier” del menú lateral izquierdo. El sistema cargará un visor con todos los reportes. Comprobar que aparecen todos los reportes de la incidencia en visor, en el mapa y en la línea temporal.

Resultado	Los reportes se consultaron correctamente.
------------------	--

Tabla 246. Prueba P-26.

P-27	
Objetivo	Comprobar que se pueden consultar los usuarios registrados en el sistema.
Procedimiento	<ul style="list-style-type: none">• Pulsar el botón “<i>Users</i>” del menú superior.• El sistema cargará una lista de usuarios registrados en el sistema.• Comprobar que aparecen todos los usuarios en la lista mostrada.
Resultado	Los usuarios se consultaron correctamente.

Tabla 247. Prueba P-27.

P-28	
Objetivo	Comprobar que se pueden consultar la ayuda del sistema.
Procedimiento	<ul style="list-style-type: none">• Pulsar el botón “<i>Help</i>” del menú superior.• El sistema cargará la ayuda del sistema.• Comprobar que la ayuda se carga correctamente.
Resultado	La ayuda no ha sido elaborada.

Tabla 248. Prueba P-28.

5.2 Análisis de resultados

Para evaluar la aplicación construida se realizará una trazabilidad de los objetivos del sistema con las pruebas realizadas. También se realizará una trazabilidad de requisitos funcionales con pruebas de evaluación y viceversa.

Análisis de los objetivos	
Dotar de información espacio/temporal visual a un conjunto de incidencias.	P-01, P-13
Diferenciar los distintos tipos de reportes asociados a cada una de las incidencias.	P-15
Posibilidad de comparar de forma visual dos conjuntos de datos utilizando su información espacio/temporal.	P-09

Tabla 249. Análisis de los objetivos.

Trazabilidad Pruebas de Evaluación – Requisitos Funcionales			
P-01	RF-05	P-15	RF-03, RF-30, RF-31
P-02	RF-06	P-16	RF-01, RF-32
P-03	RF-10, RF-14, RF-18	P-17	RF-02, RF-32
P-04	RF-10, RF-14, RF-18	P-18	RF-03, RF-32
P-05	RF-09	P-19	RF-32
P-06	RF-09	P-20	RF-32
P-07	RF-07, RF-08, RF-12, RF-13, RF-16, RF-17	P-21	RF-32
P-08	RF-07, RF-08	P-22	RF-04, RF-24, RF-25, RF-26, RF-27
P-09	RF-19	P-23	RF-04, RF-24, RF-25, RF-26, RF-27
P-10	RF-23	P-24	RF-33
P-11	RF-22	P-25	RF-11
P-12	RF-20, RF-21	P-26	RF-15
P-13	RF-01, RF-28, RF-31	P-27	RF-33
P-14	RF-02, RF-29, RF-31	P-28	RF-34

Tabla 250. Matriz de trazabilidad Pruebas-Requisitos Funcionales.

Trazabilidad Requisitos Funcionales - Pruebas de Evaluación			
RF-01	P-13, P-16	RF-18	P-03, P-04
RF-02	P-14, P-17	RF-19	P-09
RF-03	P-15, P-18	RF-20	P-12
RF-04	P-22, P-23	RF-21	P-12
RF-05	P-01	RF-22	P-11
RF-06	P-02	RF-23	P-10
RF-07	P-07, P-08	RF-24	P-22, P-23
RF-08	P-07, P-08	RF-25	P-22, P-23
RF-09	P-05, P-06	RF-26	P-22, P-23
RF-10	P-03, P-04	RF-27	P-22, P-23
RF-11	P-25	RF-28	P-13
RF-12	P-07	RF-29	P-14
RF-13	P-07	RF-30	P-15
RF-14	P-03, P-04	RF-31	P-13, P-14, P-15

RF-15	P-26	RF-32	P-16, P-17, P-18, P-19, P-20, P-21
RF-16	P-07	RF-33	P-24, P-27
RF-17	P-07	RF-34	P-28

Tabla 251. Matriz trazabilidad Requisitos Funcionales-Pruebas de Evaluación.

En la Tabla 249. Análisis de los objetivos. se puede observar que los objetivos del sistema son cubiertos con las pruebas 1, 9, 13 y 15.

Para asegurar que el sistema construido implementa todas las funcionalidades que se definieron en el diseño del sistema, se tiene la Tabla 250. Matriz de trazabilidad Pruebas-Requisitos Funcionales. donde se puede ver que todas las pruebas tienen el propósito de demostrar al menos un requisito funcional. La siguiente Tabla 251. Matriz trazabilidad Requisitos Funcionales-Pruebas de Evaluación. sirve para poder confirmar que todos los requisitos funcionales han sido implementados.

Se puede garantizar que todos los Requisitos funcionales han sido implementados, y por consiguiente se ha conseguido un sistema que cumple con todo lo que se definió en la Solución.

6 Conclusión

6.1 Aportaciones realizadas

Las aportaciones realizadas por este proyecto se refieren al ámbito de situaciones de emergencia, concretamente incendios forestales.

La principal aportación es la mejora de la representación y acceso a la información espacio-temporal de un incendio forestal. La falta de claridad y precisión a la hora de introducir o presentar la información espacio-temporal de una incidencia observada en otros sistemas, como por ejemplo SIU6, hizo la necesidad de mejorar la forma en el que un usuario consulta o almacena este tipo de datos. El análisis y estudio llevado a cabo en el capítulo Revisión de tecnologías del presente documento, ha servido para seleccionar una solución que ayude a alcanzar este objetivo. Mediante la representación espacio-temporal de cada incidencia que se ha implementado, resulta mucho más sencillo encontrar patrones de comportamiento y concentraciones de incendios forestales.

La siguiente aportación que se quiere señalar es la posibilidad de realizar comparaciones entre dos conjuntos de incidencias con el fin de detectar relaciones entre distintos tipos de incendios forestales. En principio, los dos conjuntos son iguales y cada uno de ellos cuenta con todos los incendios forestales que el sistema tiene almacenado. Aplicando los filtros que se han elaborado para cada conjunto de incidencias, se hace posible ocultar o mostrar un grupo de incendios forestales que no cumplan los requisitos de los distintos filtros, logrando construir en tiempo real dos colecciones de datos que se quieren comparar. Esta aportación permitirá ahorrar tiempo a la hora de recopilar información para realizar estudios comparativos entre dos grupos de incidencias.

Otra de las aportaciones destacables de este proyecto es la información que se puede asociar a cada incendio forestal. Dicha información se ha dividido en acciones (datos objetivos) y reportes (datos subjetivos) con el fin de separar los distintos canales que sirven como fuentes de datos.

Por último se considera también como una importante aportación la aplicación práctica de distintas tecnologías que se han utilizado en el desarrollo del proyecto. Todas estas herramientas se han utilizado en el desarrollo de un sistema que integra todas las anteriores aportaciones. Se ha procurado que el sistema construido sea robusto y adaptable a otro tipo de situaciones de emergencia. Esto se ha logrado mediante la utilización de diferentes tecnologías que hacen posible realizar cambios invirtiendo el menor tiempo posible.

6.2 Trabajos futuros

El primer trabajo futuro se refiere a la evolución o ampliación del sistema desarrollado así como a la mejora y optimización de las funciones existentes.

En primer lugar, en cuanto a la ampliación del sistema sería interesante realizar cualquiera de las siguientes aportaciones:

- **Carga de datos:** actualmente el sistema es capaz de cargar la información que existe en su base de datos y que ha sido introducida por un usuario mediante la aplicación web desarrollada. Sería interesante que existiera la posibilidad de cargar datos que no estén almacenados en el sistema mediante una fuente externa (a través de un archivo XML como una carga local o mediante una consulta a una base de datos ajena al sistema como

una carga a distancia). Esta ampliación no sería muy laboriosa ya que el sistema ha sido diseñado e implementado para que pueda tratar con incidencias genéricas.

- **Envío de datos:** el sistema construido es un consumidor de información que el usuario introduce mediante los distintos formularios. Si se implementase una carga de datos desde una fuente externa, sería importante considerar distintas funcionalidades de exportación y envío de datos.
- **Información a mostrar:** seleccionar la información de cada incidencia que se va a mostrar en el sistema. Este punto es importante para que el sistema sea capaz de leer y representar otro tipo de situaciones de emergencia cargadas desde una fuente externa. Aunque el sistema está preparado para interpretar otros tipos de situaciones de emergencia, sería interesante poder añadir o eliminar campos de información que se mostrarían mediante la interfaz. De este modo, se personalizaría la información dependiendo de la incidencia que se tiene que mostrar o leer.
- **Administración dinámica:** añadir, editar o eliminar datos de configuración del sistema para hacerlo más adaptable. Es importante para poder modificar los tipos y subtipos de incendios forestales o los distintos tipos de perfiles de usuario.
- **Generación de estadísticas:** un módulo de generación de estadísticas mediante gráficas sería importante ya que de forma automática el usuario podría realizar una serie de consultas que estuvieran predefinidas. Dado que el sistema trata con coordenadas GPS, es posible utilizar fórmulas matemáticas existentes para obtener datos estadísticos de las distintas incidencias.

Por otro lado, dado que el sistema construido pretende cubrir la pobre presentación espacio-temporal del sistema SIU6, otro de los trabajos futuros sería incluir la forma de representar la información espacio-temporal que se ha desarrollado en herramientas ya existentes que tengan esta carencia.

Por último, para asegurar que la información que el sistema recibe o transmite puede ser entendida por el propio sistema o por cualquier otro que sea externo, es importante adaptar la herramienta al estándar para comunicaciones de emergencia EDXL-DE. El estándar EDXL-DE, desarrollado por OASIS, permite que distintos sistemas sean interoperativos. De esta forma se garantiza que la información que se envía a otro sistema podrá ser interpretada de igual modo que la información que se recibe puede también serlo.

6.3 Problemas encontrados

Durante el desarrollo del proyecto se han encontrado diferentes dificultades que han retrasado las distintas tareas o provocado el cambio de algunas características o elementos del sistema.

La primera dificultad que apareció se deriva de la utilización de una serie de tecnologías de las que se tenían abundantes conocimientos teóricos pero escasos conocimientos prácticos. La principal tecnología que más problemas ha causado ha sido la utilización de JSF junto con el servidor de aplicaciones JBoss. Resultó imposible ejecutar un simple ejemplo de JSF en un servidor JBoss, en parte por la poca documentación detallada encontrada para integrar estas dos herramientas. En menor medida, al cambiar al servidor web Apache Tomcat (en este caso se encontró mucha más información), el problema apareció en forma de falta de práctica en la utilización de JSF. La agilidad a la hora de utilizar esta tecnología fue creciendo a medida que se avanzaba en la fase de implementación y crecía el contacto con JSF.

La siguiente dificultad apareció en el momento de integrar la aplicación web con Google Maps y Simile Timeline, es decir, con Timemap. En el momento de inicio del proyecto fin de carrera la versión disponible de la API de Timemap era la 1.5. Esta versión no permitía utilizar dos timemaps en la misma página de navegación y se tuvieron que revisar los archivos JavaScript de las librerías proporcionados por Timemap 1.5 para conseguir separar la información de cada timemap, ya que esta versión utiliza variables comunes que necesariamente deben ser independientes para cada timemap. Este proceso de cambio llevó bastantes horas de investigación y el resultado fue la obtención de unas librerías modificadas que permitían representar cualquier número de timemaps en una misma página de navegación. Pasado un tiempo apareció la versión Timemap 1.6 que si permitía la visualización de cualquier número de timemaps, haciendo el trabajo anterior inútil.

Por último, se menciona que la documentación utilizada asociada a la API de Timemap proporcionada por Google es caótica e imprecisa. Esta documentación es muy escasa, y la que existe está poco explicada. El problema aparece a la hora de mostrar elementos en un timemap, es decir, desplazar la línea temporal y mostrar la incidencia en el mapa. Esta tarea se basó en un método de prueba y error hasta que se consiguió averiguar cómo utilizar los objetos JavaScript que la librería de Timemap proporciona.

Además, el tiempo dedicado a conectar los diferentes filtros que se han implementado (fecha, tipo, subtipo y texto) ha sido muy elevado. En este caso, la documentación es nula y se tuvo que recurrir al estudio de filtros parecidos encontrados en varias páginas de internet. El proceso de estudio se basó en el análisis de los distintos archivos JavaScript de la API de Timemap que trataban con la información que se pretendía filtrar (ocultar o mostrar). Al final se consiguió un perfecto funcionamiento de todos los filtros implementados.

Todos estos problemas han causado retrasos en los plazos de entrega de las distintas etapas del proyecto (ver Anexo II. Seguimiento de proyecto fin de carrera).

6.4 Opiniones personales

La dificultad del proyecto no la considero alta, pero esto no quita que el sistema construido pueda ser de utilidad o pueda servir como base para otro proyecto más amplio. Es aquí cuando uno ve que ha aportado algo de utilidad y que puede ser mejorado o utilizado como base para otra herramienta.

En un principio tenía pensado utilizar un servlet para recibir todos los datos y peticiones que un usuario pudiera enviar o solicitar. Con el tiempo observé las ventajas de utilizar la tecnología JSF, ya que me ahorraría tiempo de recibir los datos, comprobar los formatos y lo que es más importante aprender a aplicar y utilizar esta tecnología, ya que nunca la había aplicado. Además, facilitó las tareas de internacionalización de la aplicación. Por consiguiente, considero útil todos los conocimientos adquiridos al utilizar las distintas tecnologías empleadas en el desarrollo del sistema, y espero que todo lo aprendido me sirva para futuros proyectos.

Por último, ha sido interesante el desarrollo del capítulo Gestión de proyecto software, ya que se ha tenido que realizar como si se tratase de un proyecto real. Esto afectó a la hora de realizar un presupuesto coherente con los recursos que creo que harían falta para el desarrollo de este proyecto. Además, este capítulo ayuda a tener en cuenta elementos que no se tendrían en mente como por ejemplo la gestión de riesgos indicada (Gestión de riesgos).

Para terminar, espero que el proyecto desarrollado y posibles ampliaciones, sean de utilidad y pueda ser utilizado para un fin mayor.

7 Bibliografía

- [1] *Ministerio de Medio Ambiente*, Estadísticas de incendios en España.
 - [2] *Sophia Liu*, “Spatiotemporal Mashups: A Survey of Current Tools to Inform Next Generation Crisis Support”.
 - [3] *Sophia Liu*, “Spatiotemporal Mashups: A Survey of Current Tools to Inform Next Generation Crisis Support”.
 - [4] *Ministerio de Medio Ambiente*, Estadísticas de incendios en España.
 - [5] Sistema SIU6. Herramienta desarrollada por el grupo DEI, Universidad Carlos III de Madrid.
 - [6] *Stephen Few*, “Data visualization past, present and future”.
 - [7] *Google Inc*, Portal web de Google Maps (2011).
 - [8] *Simile Timeline* (2011), *Massachusetts Institute of Technology*.
 - [9] *Ministerio de política territorial y administración pública*, Descripción de Métrica V3 (2010).
 - [10] *Oracle*, MySQL homepage (2011).
 - [11] *The Apache Software Foundation*
 - [12] *The Eclipse Foundation open source community website*
- Techtear.com: Blog de magazine de tecnología (2007). *Crea tu propio mashup con Google Maps (Parte 1)*. Recuperado el 15 de septiembre de 2010, de <http://www.techtear.com/2007/03/31/curso-crea-tu-propio-mashup-con-google-maps-parte-1/>
- Techtear.com: Blog de magazine de tecnología (2007). *Crea tu propio mashup con Google Maps (Parte 2)*. Recuperado el 15 de septiembre de 2010, de <http://www.techtear.com/2007/04/22/curso-crea-tu-propio-mashup-con-google-maps-parte-2/>
- Google code: Google Maps API Familiy (2010). *Google Maps AI Family*. Recuperado el 15 de septiembre de 2010, de <http://code.google.com/intl/es-ES/apis/maps/index.html>
- Maestros del web (2010). *Trabajando con el API de Google Maps*. Recuperado el 15 de septiembre de 2010, de <http://www.maestrosdelweb.com/editorial/trabajando-con-el-api-de-google-maps/>
- Adictos al trabajo (2006). *Integrando Google Maps en tus Web/Portales*. Recuperado el 15 de septiembre de 2010, de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=googlemaps>

MBnet (2010). *Google Maps API Examples and Test Pages*. Recuperado el 15 de septiembre de 2010, de <http://koti.mbnet.fi/ojalesa/exam/index.html>

Code Google (2010). *The NPR SIMILE Timeline*. Recuperado el 15 de septiembre de 2010, de <http://code.google.com/p/npr-simile-timeline/>

Dipity Blog (2009). *Simile Timeline XML Importer*. Recuperado el 15 de septiembre de 2010, de <http://blog.dipity.com/2009/02/19/simile-timeline-xml-importer/>

Simile Timeline (2010). *Timeline Examples*. Recuperado el 15 de septiembre de 2010, de <http://simile.mit.edu/timeline/examples/>

Simile Timeline (2010). *The JFK Assassination Timeline*. Recuperado el 15 de septiembre de 2010, de <http://simile.mit.edu/timeline/examples/jfk/jfk.html>

Timemap Google Code (2011). *Timemap.js Examples*. Recuperado el 15 de septiembre de 2010, de <http://timemap.googlecode.com/svn/trunk/examples/index.html>

API Timemap v.1.6 (2010). *Class TimeMapItem*. Recuperado el 15 de septiembre de 2010, de <http://opencontext.org/public/js/timemap/docs/symbols/TimeMapItem.html#constructor>

Simile Timeline (2006). *Timeline Class*. Recuperado el 15 de noviembre de 2010, de http://simile.mit.edu/wiki/Timeline_Class

Code Google (2010). *Simile Widgets*. Recuperado el 15 de septiembre de 2010, de <http://code.google.com/p/simile-widgets/wiki/Timeline>

Bird Theme (2010). *Digitizer Tool*. Recuperado el 15 de octubre de 2010, de <http://www.birdtheme.org/useful/googletool.html>

Stack Overflow (2010). *How do you create a polygon shape in a KML that is positioned in the air?* Recuperado el 15 de septiembre de 2010, de <http://stackoverflow.com/questions/2199991/how-do-you-create-a-polygon-shape-in-a-kml-that-is-positioned-in-the-air>

Simile Widgets (2009). *Religion Timelines: Jewish History + Christianity History*. Recuperado el 15 de octubre de 2010, de <http://simile-widgets.googlecode.com/svn/timeline/trunk/src/webapp/examples/religions/religions.html>

Anexo I. Control de versiones

Versión	1.0
Fecha	14 de septiembre de 2010
Descripción	<p>Contenido añadido:</p> <ul style="list-style-type: none"> • Capítulo 2, Estudio del problema: <ul style="list-style-type: none"> ○ Contexto del problema. ○ Estado de la cuestión. ○ Definición del problema. • Capítulo 4, Solución: <ul style="list-style-type: none"> ○ Descripción de la solución.

Tabla 252. Versión 1.0 Documento.

Versión	1.1
Fecha	16 de enero de 2011
Descripción	<p>Contenido añadido:</p> <ul style="list-style-type: none"> • Capítulo 4, Solución: <ul style="list-style-type: none"> ○ Análisis. ○ Diseño de sistema. <p>Cambios realizados:</p> <ul style="list-style-type: none"> • Capítulo 2, Estudio del problema: <ul style="list-style-type: none"> ○ Contexto del problema. ○ Estado de la cuestión. ○ Definición del problema. • Capítulo 4, Solución: <ul style="list-style-type: none"> ○ Descripción de la solución.

Tabla 253. Versión 1.1 Documento.

Versión	1.2
Fecha	19 de enero de 2011
Descripción	<p>Contenido añadido:</p> <ul style="list-style-type: none"> • Capítulo 4, Solución: <ul style="list-style-type: none"> ○ Implementación. ○ Producto de desarrollo.

Tabla 254. Versión 1.2 Documento.

Versión	1.3
Fecha	24 de enero de 2011
Descripción	<p>Contenido añadido:</p> <ul style="list-style-type: none"> • Capítulo 3, Gestión de proyecto software: <ul style="list-style-type: none"> ○ Alcance del proyecto. ○ Plan de trabajo. ○ Gestión de riesgos. ○ Plan de pruebas.

Tabla 255. Versión 1.3 Documento.

Versión	1.4
Fecha	17 de febrero de 2011
Descripción	<p>Contenido añadido:</p> <ul style="list-style-type: none"> • Glosario de términos: añadidos todos los términos que requieren una pequeña definición. • Capítulo 1, Introducción: <ul style="list-style-type: none"> ○ Inclusión de los apartados Planteamiento del problema, Objetivos, Metodología, Estructura de la Memoria. • Capítulo 4, Solución: <ul style="list-style-type: none"> ○ Pruebas: pruebas caja blanca, pruebas caja negra y pruebas de integración. • Capítulo 5, Evaluación: <ul style="list-style-type: none"> ○ Proceso de evaluación: plan de pruebas y casos de prueba realizados. ○ Análisis de resultados: análisis realizado sobre el conjunto de pruebas. • Capítulo 7, Bibliografía: inclusión de todas las fuentes consultadas para la realización del proyecto. • Anexo I, Control de versiones: recopilación del contenido y cambios realizados en todas las versiones entregadas. • Anexo III, Manual de Usuario: creación del contenido del manual de usuario recogiendo todas las funciones del sistema construido. • Anexo IV, Prototipo: pantallazos más relevantes de la aplicación. <p>Cambios realizados:</p> <ul style="list-style-type: none"> • Capítulo 2, Estudio del problema: <ul style="list-style-type: none"> ○ Renombrado del capítulo a Revisión de Tecnologías. ○ Eliminación apartado Herramientas Existentes. • Anexo I, Control de versiones: inclusión de la versión 1.4 de la memoria.

Tabla 256. Versión 1.4 Documento.

Versión	1.5
Fecha	28 de febrero de 2011
Descripción	<p>Contenido añadido:</p> <ul style="list-style-type: none"> • Portada del documento: añadidos los campos título, subtítulo y versión del documento. • Resumen del documento. • Apartado 3.2, Gestión de recursos. • Capítulo 6, Conclusión: <ul style="list-style-type: none"> ○ Inclusión de los apartados Aportaciones realizadas, Trabajos futuros, problemas encontrados y opiniones personales. <p>Cambios realizados:</p> <ul style="list-style-type: none"> • Capítulo 7, Bibliografía: <ul style="list-style-type: none"> ○ Adecuación de las entradas al formato APA. • Anexo I, Control de versiones: inclusión de la versión 1.5 de la memoria.

Tabla 257. Versión 1.5 Documento.

Versión	1.6
Fecha	14 de abril de 2011
Descripción	Contenido modificado: <ul style="list-style-type: none">• Anexo II, Seguimiento de proyecto fin de carrera.

Tabla 258. Versión 1.6 Documento.

Versión	1.7
Fecha	2 de mayo de 2011
Descripción	Contenido modificado: <ul style="list-style-type: none">• Título de PFC.

Tabla 259. Versión 1.7 Documento.

Anexo II. Seguimiento de proyecto fin de carrera

Forma de seguimiento

La forma de realizar el seguimiento de este proyecto fin de carrera consistirá en la medida de tres valores. Estos son:

- Planificación inicial.
- Re-planificación.
- Planificación real.

Con estos datos se podrá saber de una forma rápida el tiempo invertido en la realización del proyecto, los cambios o modificaciones de fechas, los retrasos sufridos, etc.

Planificación inicial

TAREAS	FECHA INICIO	FECHA FIN
Memoria	01/07/2009	20/03/2010
Introducción	01/02/2010	10/02/2010
Estudio del problema	01/07/2009	10/07/2009
Gestión del proyecto software	15/09/2009	30/09/2009
Solución	10/07/2009	31/08/2009
Evaluación	08/01/2010	31/01/2010
Conclusión	10/02/2010	28/02/2010
Anexos	01/03/2010	20/03/2010
Estudio inicial	01/06/2009	31/07/2009
Estudio del problema	01/06/2009	01/07/2009
Esbozo de solución	01/07/2009	31/07/2009
Desarrollo	10/07/2009	31/01/2010
Definición de requisitos	10/07/2009	20/07/2009
Especificación de requisitos	20/07/2009	10/08/2009
Diseño	10/08/2009	31/08/2009
Diseño interfaz	10/08/2009	25/08/2009
Diseño modelo de datos	10/08/2009	25/08/2009
Diseño lógica de negocio	15/08/2009	31/08/2009
Implementación	01/09/2009	22/12/2009
Pruebas	08/01/2010	31/01/2010
Defensa	Junio 2010	Junio 2010

Tabla 260. Planificación inicial del PFC.

Planificación final

FECHA FIN	FECHA INICIO	FECHA FIN	TIEMPO
Memoria	10/01/2011	02/05/2011	230
Introducción	28/02/2011	01/03/2011	6
Estudio del problema	10/01/2011	16/01/2011	44
Gestión del proyecto software	21/02/2011	27/02/2011	17
Solución	16/01/2011	20/02/2011	146
Evaluación	23/02/2011	27/02/2011	6
Conclusión	28/02/2011	01/03/2011	4

Anexos	28/02/2011	01/03/2011	7
Estudio inicial	01/06/2010	12/09/2010	56
Estudio del problema	01/06/2010	01/09/2010	30
Esbozo de solución	02/09/2010	12/09/2010	26
Desarrollo	13/09/2010	17/12/2010	264
Definición de requisitos	13/09/2010	19/09/2010	6
Especificación de requisitos	20/09/2010	03/10/2010	10
Diseño	04/10/2010	17/10/2010	32
Diseño interfaz	04/10/2010	10/10/2010	18
Diseño modelo de datos	11/10/2010	12/10/2010	10
Diseño lógica de negocio	13/10/2010	17/10/2010	12
Implementación	15/09/2010	15/12/2010	210
Pruebas	16/12/2010	17/12/2010	6
Defensa	Abril 2011	Abril 2011	16
Presentación	10/04/2011	2/05/2011	16

Tabla 261. Planificación real del PFC.

Anexo III. Manual de usuario

Acceso al sistema



Figura 30. Pantalla de acceso al sistema.

Para acceder al sistema el usuario debe introducir su nombre de usuario en el campo *User* y su contraseña en el campo *Password*. A continuación debe pulsar sobre el botón *Log in*.

Una vez que el usuario ha accedido al sistema aparecerá la siguiente ventana:

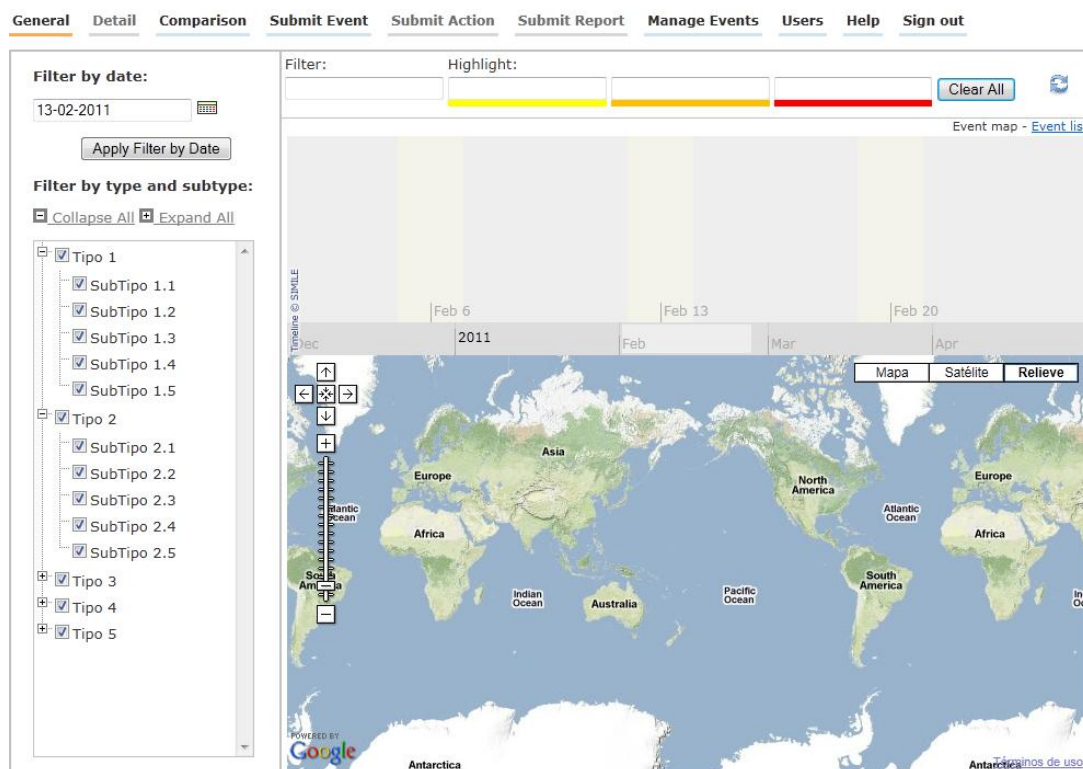


Figura 31. Pantalla general del sistema.

Menú superior

El menú contiene el acceso a las principales acciones del sistema. Estas son:

- Vistas: permite consultar la vista general, detalle y comparación.

General Detail Comparison

Figura 32. Menú Vistas.

- Añadir elementos: permite añadir incidencias, acciones y reportes.

Submit Event Submit Action Submit Report

Figura 33. Menú Añadir.

- Administración de elementos: permite editar o eliminar incidencias, acciones y reportes.

Manage Events

Figura 34. Menú Administración de Elementos.

- Administración de usuarios: permite editar, eliminar y añadir usuarios.

Users

Figura 35. Menú Administración de Usuarios.

- Ayuda: permite consultar la ayuda del sistema.

Help

Figura 36. Menú Ayuda.

- Salir del sistema: permite salir del sistema.

Sign out

Figura 37. Menú Salir del Sistema.

La barra inferior de color naranja (**Figura 38. Menú Seleccionado.**) indica que el usuario está situado en dicha opción del sistema. La barra de color gris bajo el botón (**Figura 39. Menú Deshabilitado.**) indica que el botón está deshabilitado.

Submit Event Submit Action Submit Report

Figura 38. Menú Seleccionado.

General Detail Comparison

Figura 39. Menú Deshabilitado.

Vista General

La vista General contiene a todas las incidencias ubicadas en un mapa y una línea temporal o en una lista. Esta vista sirve para localizar cualquier incidencia almacenada en el sistema.

Para seleccionar el mapa de la vista general habrá que pulsar sobre el botón *Event Map* y sobre el botón *Event List* para cargar la lista.

El mapa de la vista general es el siguiente:

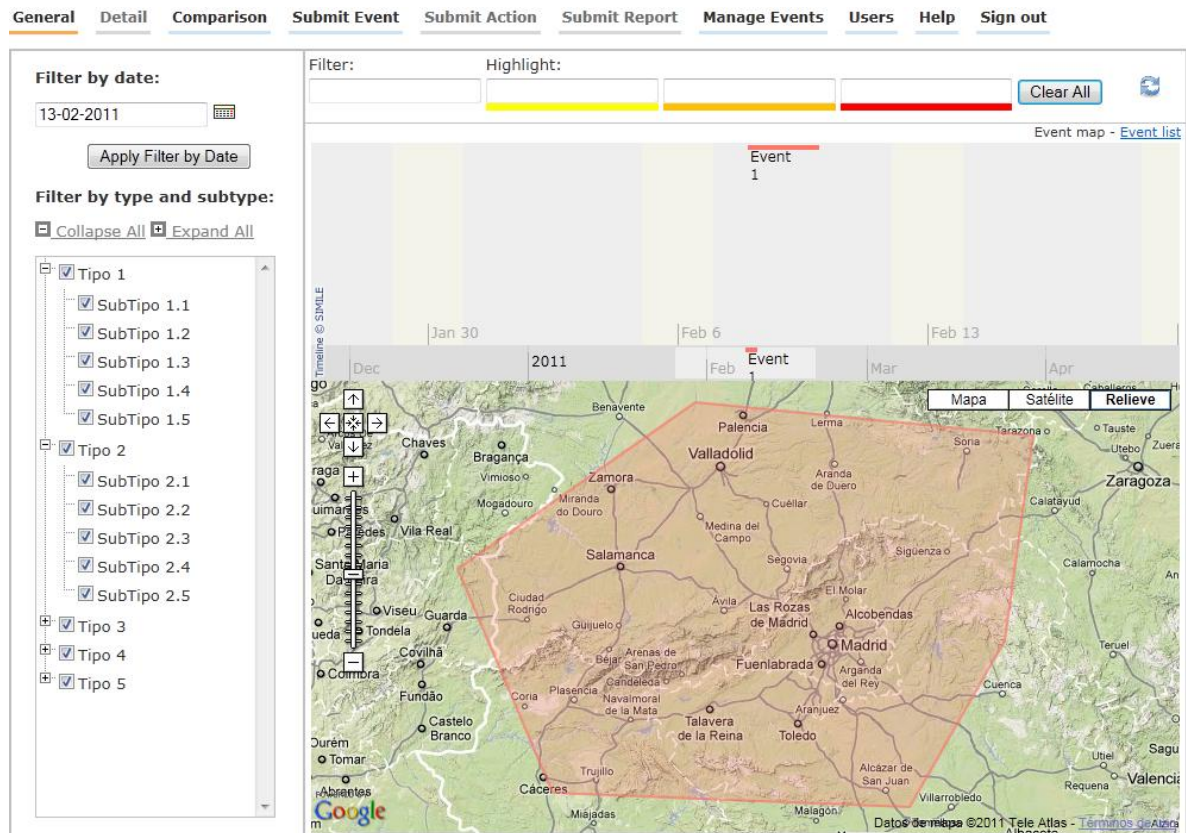


Figura 40. Mapa Vista General.

Se destacan cuatro elementos de la pantalla:

- Filtro por fecha y filtro por tipos y subtipos:
 - El filtro por fecha ubica la línea temporal en la fecha introducida.
 - El filtro por tipos y subtipos oculta o muestra las incidencias que tengan los tipos o subtipos marcados o desmarcados.
- Filtro por texto:
 - El filtro por texto oculta las incidencias que no tengan el texto introducido en el título.
 - El highlight resalta las incidencias que tengan el texto introducido en el título.
- Línea temporal: sitúa en el tiempo a las incidencias.
- Mapa: sitúa en el espacio a las incidencias.

La lista de la vista general es el siguiente:

General Detail Comparison Submit Event Submit Action Submit Report Manage Events Users Help Sign out

Filter by date:

Filter by type and subtype:

☐ Collapse All ☐ Expand All

- ☒ Tipo 1
 - ☒ SubTipo 1.1
 - ☒ SubTipo 1.2
 - ☒ SubTipo 1.3
 - ☒ SubTipo 1.4
 - ☒ SubTipo 1.5
- ☒ Tipo 2
 - ☒ SubTipo 2.1
 - ☒ SubTipo 2.2
 - ☒ SubTipo 2.3
 - ☒ SubTipo 2.4
 - ☒ SubTipo 2.5
- ☒ Tipo 3
- ☒ Tipo 4
- ☒ Tipo 5

Filter:

[Event map](#) - [Event list](#)

Title	Type	Subtype	From	To	Actions
Event 1	Tipo 1	SubTipo 1.1	08-02-2011	10-02-2011	Detail

Figura 41. Lista Vista General.

Se destacan tres elementos de la pantalla:

- Filtro por fecha y filtro por tipos y subtipos:
 - El filtro por fecha ubica la línea temporal en la fecha introducida.
 - El filtro por tipos y subtipos oculta o muestra las incidencias que tengan los tipos o subtipos marcados o desmarcados.
- Filtro por texto:
 - El filtro por texto resalta las incidencias que contengan en el título el texto introducido.
- Lista con todas las incidencias.

Desde la vista general, ya sea desde el mapa o lista, se accede a la vista de detalle de cada incidencia. Desde el mapa hay que pinchar sobre la incidencia y seguidamente sobre el botón *Detail* (**Figura 42**). Desde la lista hay que pinchar sobre el botón *Detail* situado en la columna de las acciones (**Figura 43**).

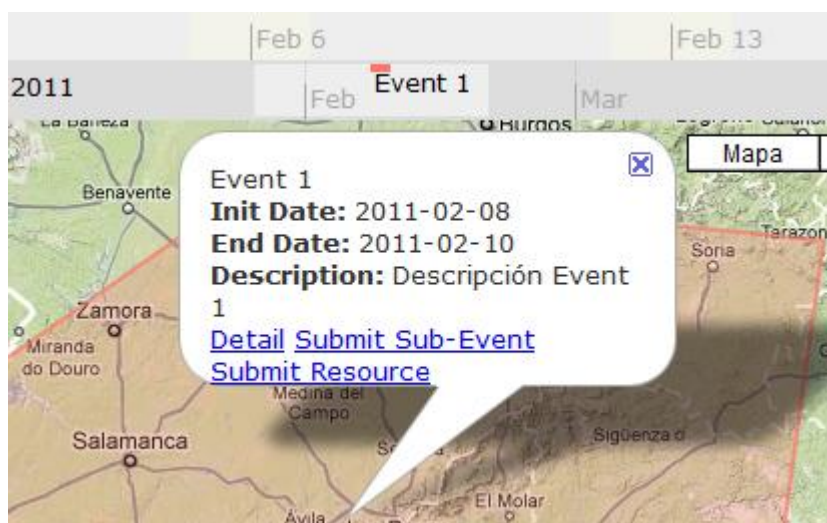


Figura 42. Globo Mapa de la Vista General.

Event map - Event list		
From	To	Actions
08-02-2011	10-02-2011	Detail

Figura 43. Columna de Acciones.

Vista Detalle

La vista en detalle muestra todas las acciones y reportes de una incidencia. Para acceder a la vista en detalle hay que pulsar sobre el botón *Detail* (**Figura 42** y **Figura 43**).

Una vez accedido el sistema cargará por defecto la pantalla de detalle de acciones de la incidencia (**Figura 44**). En esta pantalla aparecerán todas las acciones que se hayan asociado a la incidencia. Esta pantalla se divide en cuatro zonas:

- Información de la incidencia a la que pertenecen todas las acciones.
- Filtros de fecha y texto.
- Mapa y línea temporal en el que aparecen todas las acciones.
- Lista de todas las acciones.

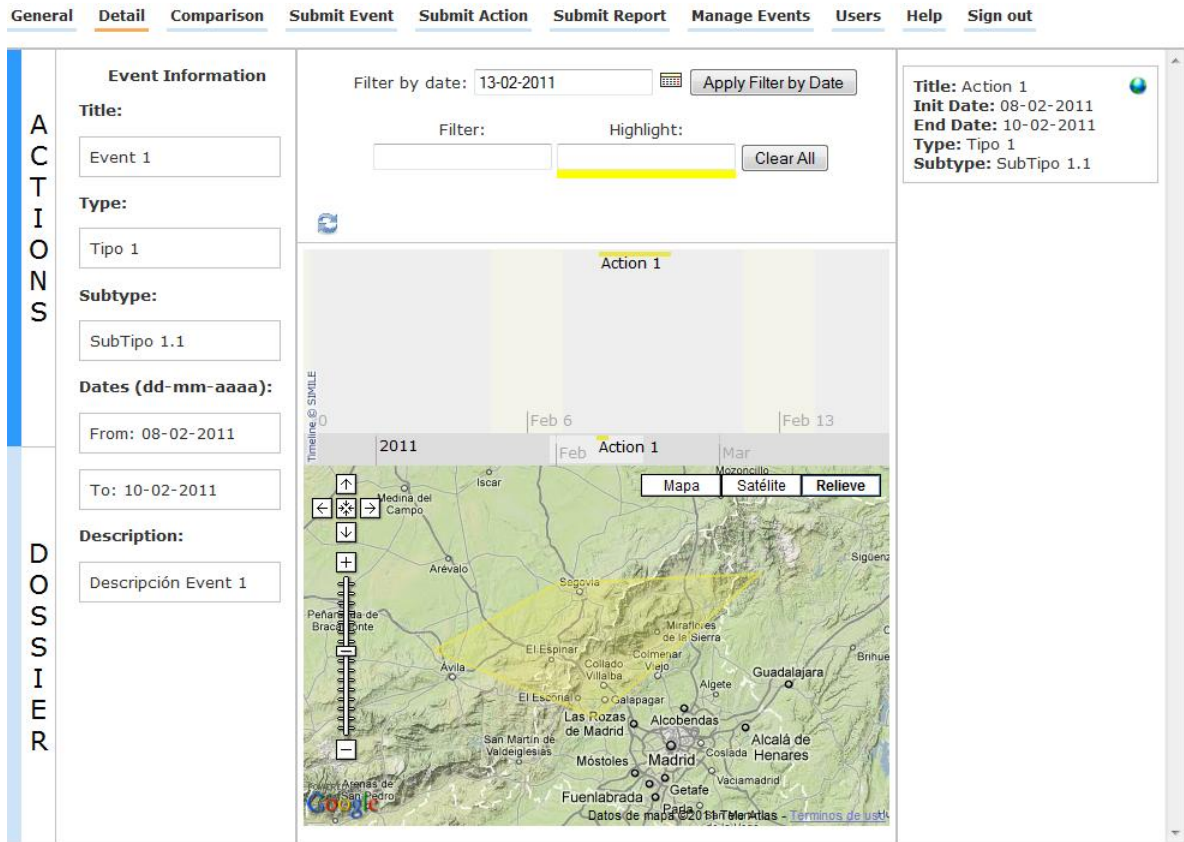


Figura 44. Vista en Detalle - Acciones.

Para acceder a la vista de detalle de reportes hay que pulsar sobre el botón *Dossier* situado a la izquierda. En esta pantalla (**Figura 45**) aparecerán todos los reportes asociados a la incidencia. Esta pantalla se divide en cinco zonas:

- Información de la incidencia a la que pertenecen todas las acciones.
- Visor o lista en el que aparecen todos los reportes.
- Filtros de fecha y texto.
- Mapa y línea temporal en el que aparecen todos los reportes.
- Información del reporte seleccionado.

Para descargar un reporte habrá que seleccionarlo y pulsar sobre el botón *Download Report*. En el visor de la **Figura 45** aparecen (en este orden) un imagen, un documento, un vídeo, un audio y un reporte de tipo otro. Las imágenes aparecen pre visualizadas en miniatura.

Figura 45. Dossier Vista de Detalle.

Vista Comparación

La vista de comparación muestra dos mapas con todas las incidencias. Para acceder a la vista de comparación hay que pulsar sobre el botón *Comparison* del menú superior (**Figura 32**).

Una vez accedido el sistema cargará la ventana con los dos mapas (**Figura 46**). Esta ventana se divide en cuatro zonas:

- Filtros por tipo y subtipos.
- Filtros de fecha.
- Filtros por texto.
- Mapas y líneas temporales independientes.

Esta vista sirve para comparar dos conjuntos de datos utilizando los diferentes filtros proporcionados.

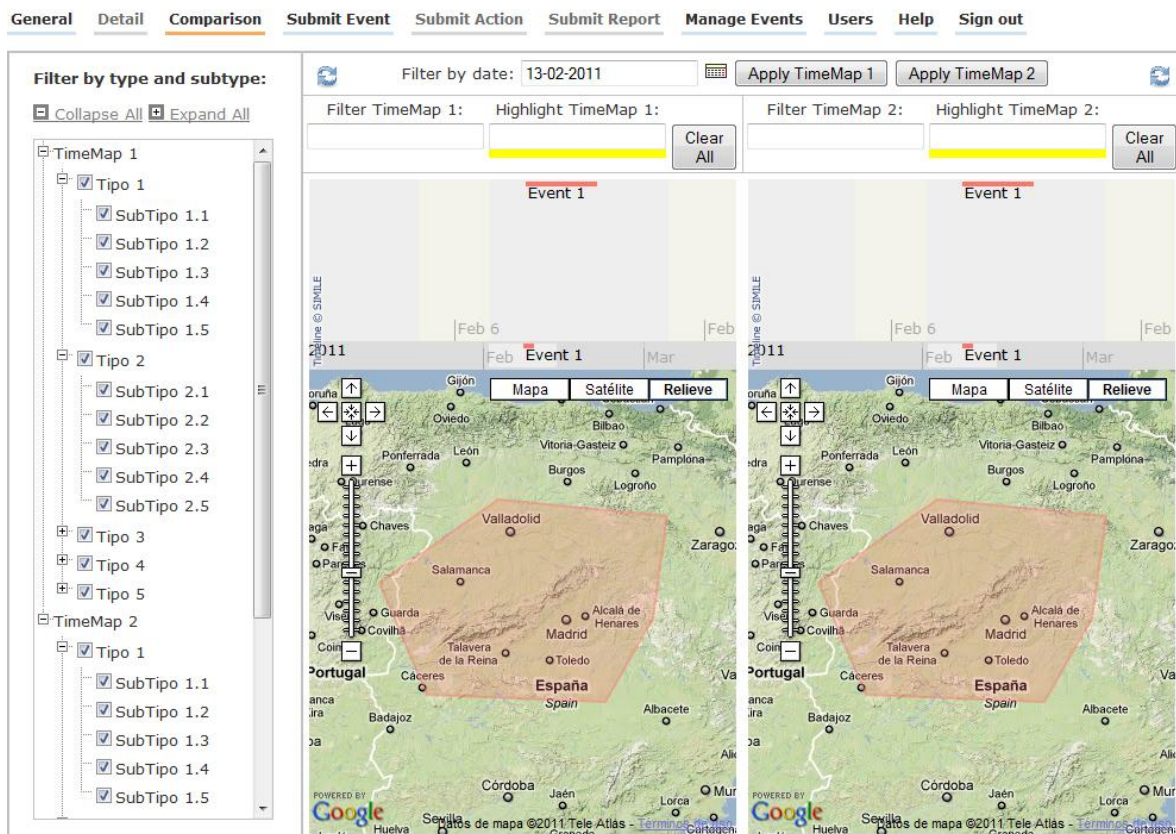


Figura 46. Vista de Comparación.

Añadir Incidencia

Las incidencias son el elemento esencial del sistema, y tienen asociadas acciones y reportes. Para añadir una incidencia hay que pulsar sobre el botón *Submit Event* del menú superior (**Figura 33. Menú Añadir.**). El sistema cargará la siguiente ventana:

The screenshot shows the 'Añadir Incidencia' form. The top navigation bar includes links: General, Detail, Comparison, **Submit Event**, Submit Action, Submit Report, Manage Events, Users, Help, and Sign out. The 'Event Information' section on the left contains the following fields:

- Title:** A text input field.
- Type:** A dropdown menu with 'Tipo 1' selected.
- Subtype:** A dropdown menu with 'SubTipo 1.1' selected.
- Dates (dd-mm-aaaa):** Two date pickers. 'From' is set to '13-02-2011' and 'To' is empty.
- Description:** A large text area.
- Buttons: 'Clear' and 'Save Event'.

The right section features a map of Europe and North Africa. Above the map, there is a 'Draw mode' dropdown set to 'Polyline' and a 'Longitude-Latitude' input field showing '45.767523, 12.172852'. Below these are buttons for 'Clear Map', 'Delete Last Point', and 'Close Polyshape'. The map itself includes a compass, a zoom slider, and a scale bar (200 mi / 200 km). A 'Mapa' dropdown menu is visible in the top right corner of the map area.

Figura 47. Pantalla

Esta pantalla contiene todos los campos necesarios para dar valores a la incidencia. Los valores que el usuario debe introducir son los siguientes:

- Título: título de la incidencia.
- Tipo: tipo de incidencia.
- Subtipo: subtipo de incidencia.
- Fecha de inicio: fecha de inicio de la incidencia.
- Fecha de finalización: fecha en la que finalizó la incidencia.
- Descripción: descripción de la incidencia.

Para introducir la ubicación consultar la sección **Añadir Ubicación**. Los campos fecha de finalización y descripción son opcionales y no es necesario darles un valor.

Si se quiere limpiar todos los campos de la incidencia se tiene que pulsar sobre el botón *Clear*. Por último el usuario debe pulsar sobre el botón *Save Event* para almacenar la incidencia. Una vez guardada, el sistema cargará la **Vista General** mostrando todas las incidencias.

Añadir Acción

Para añadir una acción hay que pulsar sobre el botón *Submit Action* del menú superior (**Figura 33. Menú Añadir.**). Si el botón está deshabilitado habrá que acceder primero a la **Vista Detalle** de la incidencia a la que se añadirá la acción. El sistema cargará la siguiente ventana:

The screenshot shows the 'Añadir Acción' (Add Action) form. At the top, there is a navigation menu with options: General, Detail, Comparison, Submit Event, Submit Action (highlighted), Submit Report, Manage Events, Users, Help, and Sign out. The form is divided into three main sections:

- Event Information:** Contains fields for Title (Event 1), Type (Tipo 1), Subtype (SubTipo 1.1), Dates (dd-mm-aaaa) (From: 08-02-2011, To: 10-02-2011), and Description (Descripción Event 1).
- Action Information:** Contains fields for Title, Type (Tipo 1), Subtype (SubTipo 1.1), Dates (dd-mm-aaaa) (From: 13-02-2011, To:), and Description.
- Map:** A central map showing a view of Spain. Above the map, there is a 'Draw mode: Polyline' dropdown and a 'Longitude-Latitude:' input field. Below the map, there are buttons for 'Clear Map', 'Delete Last Point', and 'Close Polyshape'. The map itself has a 'Mapa' dropdown menu and a scale bar indicating 200 km.

Figura 48. Añadir Acción.

La pantalla de la **Figura 48** se divide en dos zonas. La columna de la izquierda contiene la información de la incidencia a la que se va a asociar la acción. El resto de la pantalla se utilizará para dar valor a todos los campos de la acción. Los datos que el usuario tiene que introducir son:

- Título: título descriptivo de la acción.
- Tipo: tipo de la acción.
- Subtipo: subtipo de la acción.
- Fecha de inicio: fecha de inicio de la acción.
- Fecha de finalización: fecha de finalización de la acción.
- Descripción: descripción de la acción.

Para introducir la ubicación de la acción consultar la sección **Añadir Ubicación**. Los campos fecha de finalización y descripción son opcionales y no es necesario darles un valor.

Si se quiere limpiar todos los campos de la acción se tiene que pulsar sobre el botón *Clear*. Por último el usuario debe pulsar sobre el botón *Save Event* para almacenar la acción. Una vez guardada, el sistema cargará la **Vista Detalle** mostrando todas las acciones de la incidencia.

Añadir Reporte

Para añadir un reporte hay que pulsar sobre el botón *Submit Report* del menú superior (**Figura 33. Menú Añadir.**). Si el botón está deshabilitado habrá que acceder primero a la **Vista Detalle** de la incidencia a la que se añadirá el reporte. El sistema cargará la siguiente ventana:

Figura 49. Añadir Reporte.

La pantalla de la **Figura 49** se divide en dos zonas. La columna de la izquierda contiene la información de la incidencia a la que se va a asociar el reporte. El resto de la pantalla se utilizará para dar valor a todos los campos del reporte. Los datos que el usuario tiene que introducir son:

- Título: título descriptivo del reporte.
- Tipo: tipo de reporte. Los tipos disponibles son los siguientes:
 - Imagen: indica que el reporte es una imagen.
 - Video: indica que el reporte es un video.
 - Documento: indica que el reporte es un documento.
 - Audio: indica que el reporte es un audio.
 - Otro: cualquier otro tipo de reporte.
- Subtipo: subtipo de la acción.
- Fecha de inicio: fecha de inicio de la acción.
- Fecha de finalización: fecha de finalización de la acción.
- Descripción: descripción de la acción.

Para introducir la ubicación del reporte consultar la sección **Añadir Ubicación**. El campo descripción es opcional y no es necesario darle un valor.

Si se quiere limpiar todos los campos del reporte se tiene que pulsar sobre el botón *Clear*. Por último el usuario debe pulsar sobre el botón *Save Resource* para almacenar el reporte. Una vez guardado, el sistema cargará la **Vista Detalle** mostrando todos los reportes de la incidencia.

Añadir Ubicación

La ubicación es la información espacial de una incidencia, acción o reporte. La ubicación que se le puede asignar a una incidencia o acción puede ser un área o un punto. Sin embargo, la ubicación que se le puede asignar a un reporte sólo puede ser un punto.

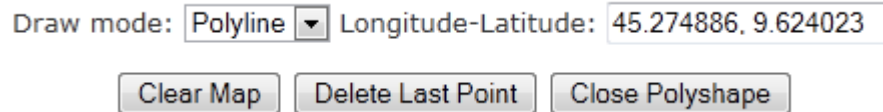


Figura 50. Controles Ubicación de Incidencia o Acción.

Atendiendo a la **Figura 50**, los controles para añadir la ubicación a una incidencia o acción son los siguientes:

- Modo de dibujo: son tres.
 - Polígono sin relleno: permite representar un polígono sin relleno.
 - Polígono con relleno: permite representar un polígono con relleno.
 - Punto: permite representar un punto.
- Longitud-Latitud: indica las coordenadas en las que se encuentra el puntero del ratón.
- Botón *Clear Map*: permite borrar la ubicación actual.
- Botón *Delete Last Point*: permite borrar el último punto introducido.
- Botón *Close Polyshape*: permite cerrar el polígono y confirmar la figura introducida. Siempre hay que pulsar este botón para confirmar la ubicación.

Para representar la ubicación hay que utilizar el mapa proporcionado e ir haciendo clic en el lugar en el que se quiera ubicar cada vértice del área (**Figura 51**). En el caso de querer representar un solo punto habrá que seleccionar el modo de dibujo *Punto* y hacer un solo clic en el lugar donde se ubicará el punto (**Figura 54**). Una vez introducida la ubicación hay que hacer clic sobre el botón *Close Polyshape* para que el polígono se cierre (**Figura 52**) y se confirme la ubicación, tanto el punto como el polígono.



Figura 51. Polígono sin cerrar ni confirmar.



Figura 52. Polígono cerrado y confirmado.



Figura 53. Polígono relleno.



Figura 54. Punto sin confirmar.

Administración de incidencias, acciones y reportes

Para editar o eliminar una incidencia, acción o reporte hay que pulsar sobre el botón *Manage Events* del menú superior (Figura 34Figura 33. Menú Añadir.). El sistema cargará la siguiente ventana:

General

Detail

Comparison

Submit Event

Submit Action

Submit Report

Manage Events

Users

Help

Sign out

Event map - Event list

Event List

Id	Title	Type	Subtype	From	Actions
2	Event 1	Tipo 1	SubTipo 1.1	08-02-2011	Delete Event Edit Event

Actions List

Id	Main Id	Title	Type	Subtype	From	Actions
4	2	Action 1	Tipo 1	SubTipo 1.1	08-02-2011	Delete Action Edit Action

Reports List

Id	Main Id	Title	Type	From	Actions
2	2	Imagen 1	Image	08-02-2011	Delete Report Edit Report
3	2	PDF	Document	08-02-2011	Delete Report Edit Report
4	2	Video	Video	08-02-2011	Delete Report Edit Report
5	2	Audio	Audio	08-02-2011	Delete Report Edit Report
6	2	Otro	Other	08-02-2011	Delete Report Edit Report

Figura 55. Administración de Incidencias, Acciones y Reportes.

En esta pantalla (Figura 55) aparecen tres listas. La primera lista contiene a todas las incidencias, la segunda lista contiene todas las acciones y la tercera lista contiene todos los reportes.

Para editar un elemento hay que pulsar sobre el botón *Edit Event*, *Edit Action* o *Edit Report*. Para eliminar un elemento hay que pulsar sobre el botón *Delete Event*, *Delete Action* o *Delete Report*.

La pantalla de edición contiene los mismos campos que aparecen en las distintas pantallas en las que se añaden incidencias, acciones o reportes.

Administración de Usuarios

Para crear, editar o eliminar un usuario hay que pulsar sobre el botón *Users* del menú superior (Figura 35Figura 33. Menú Añadir.). El sistema cargará la siguiente ventana:

General	Detail	Comparison	Submit Event	Submit Action	Submit Report	Manage Events	Users	Help	Sign out
User List								Submit User	
User	Profile	Name	Surname	Phone number	E-mail			Actions	
sigi	Perfil 1	Javier	Prieto Canela	123456789	100047539@alumnos.uc3m.es			Delete User Edit User	

Figura 56. Administración de Usuarios.

En esta pantalla (Figura 56) aparece una lista con todos los usuarios y datos existentes. Para crear un nuevo usuario hay que pulsar sobre el botón *Submit User* situado en sobre la columna *Actions* de la tabla y aparecerá la siguiente pantalla:

General	Detail	Comparison	Submit Event	Submit Action	Submit Report	Manage Events	Users	Help	Sign out
-------------------------	------------------------	----------------------------	------------------------------	-------------------------------	-------------------------------	-------------------------------	-----------------------	----------------------	--------------------------

User Information

Login Information		
User:	Password:	Profile:
<input type="text"/>	<input type="password"/>	Perfil 1 ▾

Personal Data	
Name:	Surname:
<input type="text"/>	<input type="text"/>

Phone number:	E-mail:
<input type="text"/>	<input type="text"/>

Figura 57. Añadir Usuario.

Los datos que hay rellenar son los siguientes:

- Usuario: nombre único que identifica al usuario. Utilizado para acceder al sistema.
- Contraseña: contraseña de acceso al sistema.
- Perfil: perfil de usuario asignado.
- Nombre: nombre del usuario.
- Apellidos: apellidos del usuario.
- Teléfono: teléfono del usuario.
- E-mail: correo electrónico del usuario.

Para guardar el usuario hay que pulsar sobre el botón *Save User*. El sistema avisará en el caso de que ya exista otro usuario que esté utilizando el nombre de usuario introducido. Para borrar los campos hay que pulsar sobre el botón *Clear*.

Para editar un usuario hay que pulsar sobre el botón *Edit User* de la columna *Actions*. Aparecerá la ventana mostrada en la **Figura 57**, la misma que se utiliza para crear un nuevo usuario.

Ayuda

Para acceder a la ayuda del sistema hay que pulsar sobre el botón *Help* del menú superior (**Figura 36**Figura 33. Menú Añadir.).

Salir del Sistema

Para salir del sistema hay que pulsar sobre el botón *Sign Out* del menú superior (**Figura 37**Figura 33. Menú Añadir.). Una vez pulsado el sistema cargará la pantalla de acceso al sistema (**Figura 30**).

Anexo IV. Prototipo

En este apartado se incluyen los pantallazos más relevantes sobre el contenido de la aplicación web realizada.

Pantalla de Login



Figura 58. Pantalla de Login.

Esta pantalla es la primera en mostrarse, en ella el usuario debe identificarse para poder acceder al contenido.

General

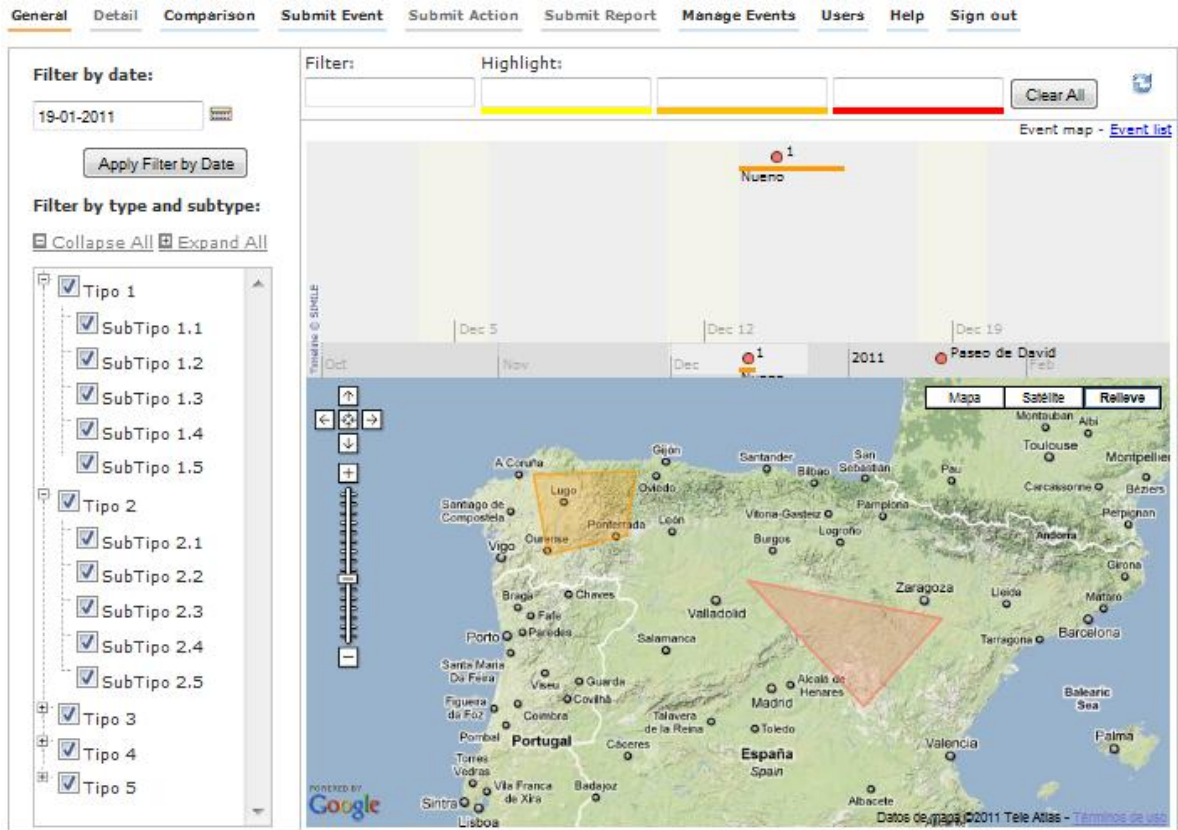


Figura 59. Pantalla General.

Esta pantalla se carga una vez que el usuario se ha identificado. En ella aparecen todas las incidencias y sirve para localizar y acceder a la información propia de cada una.

Para facilitar la tarea de búsqueda al usuario existen cuatro tipos de filtros:

- Filtro por fecha.
- Filtro por tipo y subtipo.
- Filtro de texto (sólo muestra las incidencias que coinciden con el texto introducido).
- Filtro highlight (muestra la coincidencia con color amarillo, naranja o rojo).

Comparación

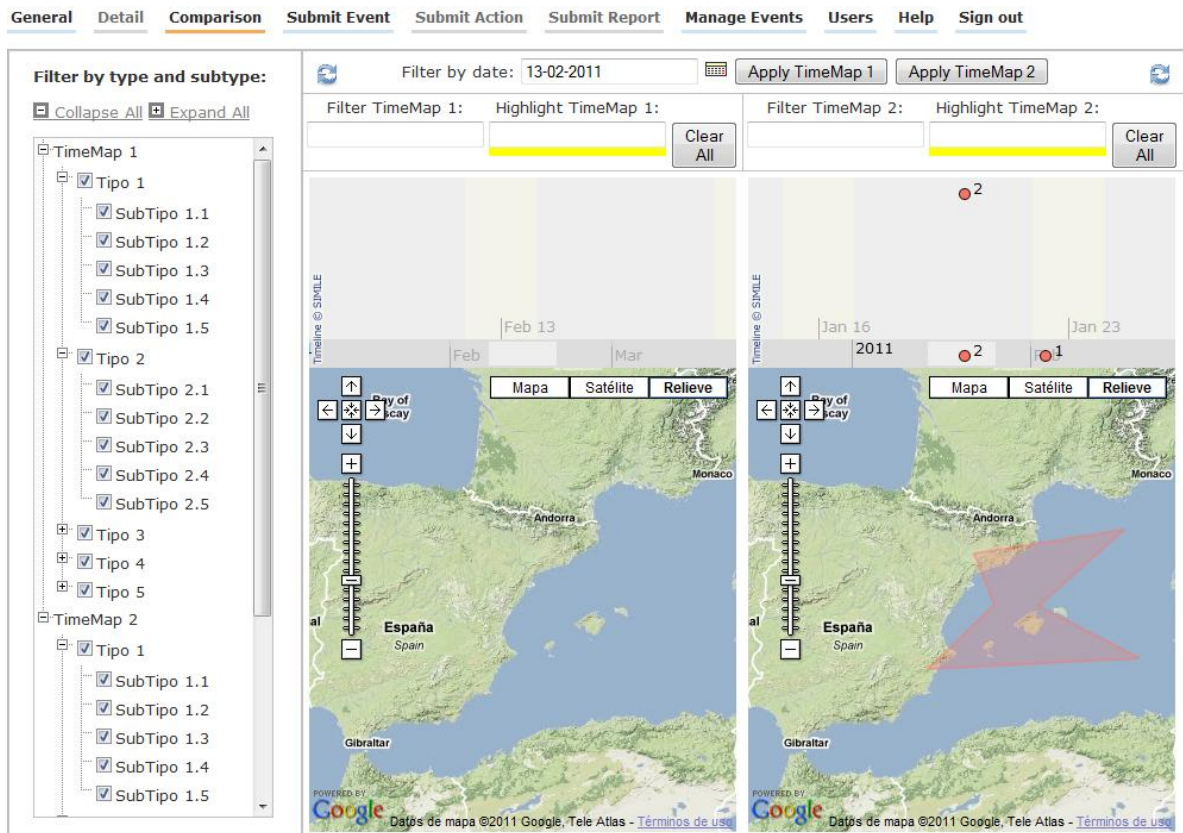


Figura 60. Pantalla Comparación.

Esta sección de la aplicación web sirve para comparar dos conjuntos de incidencias. Cada mapa representa a un conjunto de datos y cada mapa tiene un conjunto de filtros:

- Filtro por fecha.
- Filtro por tipo y subtipo.
- Filtro de texto (sólo muestra las incidencias que coinciden con el texto introducido).
- Filtro highlight (muestra la coincidencia con color amarillo).

Guardar Incidencia

General **Detail** **Comparison** **Submit Event** **Submit Action** **Submit Report** **Manage Events** **Users** **Help** **Sign out**

Event Information

Title:

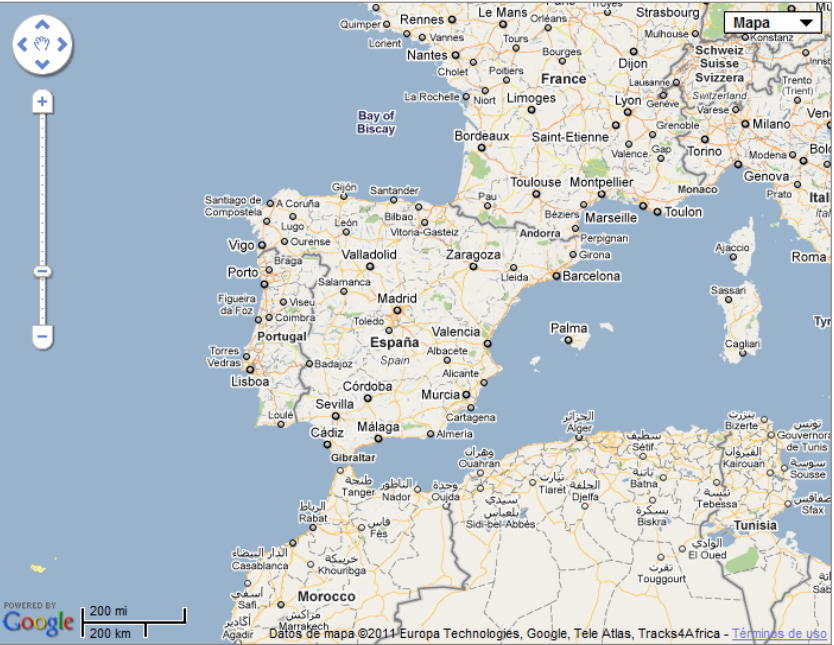
Type:
Tipo 1 ▾

Subtype:
SubTipo 1.1 ▾

Dates (dd-mm-aaaa):
From: 13-02-2011
To:

Description:

Draw mode: Polyline ▾ Longitude-Latitude: 45.767523, 12.172852



Mapa

POWERED BY Google

200 mi
200 km

Datos de mapa ©2011 Europa Technologies, Google, Tele Atlas, Tracks4Africa - [Términos de uso](#)

Figura 61. Pantalla Guardar Incidencia.

Esta pantalla se utiliza para guardar nuevas incidencias en el sistema. En la columna de la izquierda se indica el título, tipo, subtipo, fechas y descripción. En el mapa se indica la ubicación de la incidencia, que podrá ser un punto o un área.

Guardar Acción

General Detail Comparison Submit Event **Submit Action** Submit Report Manage Events Users Help Sign out

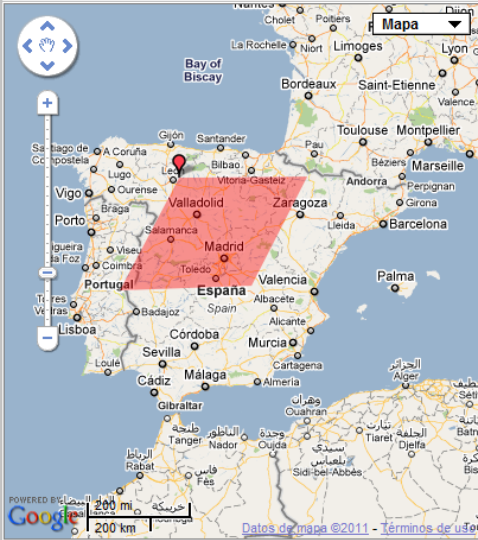
Event Information
Title:

Type:

Subtype:

Dates (dd-mm-aaaa):
From:
To:
Description:

Draw mode: Longitude-Latitude:



Action Information
Title:

Type:

Subtype:

Dates (dd-mm-aaaa):
From:
To:
Description:

Figura 62. Pantalla Guardar Acción.

Esta pantalla se utiliza para guardar una acción asociada a una incidencia. Cada acción debe tener un título, tipo, subtipo, fechas y descripción (columna derecha). También es necesario indicar la ubicación de la acción, y podrá ser un punto o un área.

En la columna izquierda aparecen los datos de la incidencia a la que se asocia la acción.

Guardar Reporte

General	Detail	Comparison	Submit Event	Submit Action	Submit Report	Manage Events	Users	Help	Sign out
<div> <div> Information Event <p>Title:</p> <input type="text" value="2"/> <p>Type:</p> <input type="text" value="Tipo 1"/> <p>Subtype:</p> <input type="text" value="SubTipo 1.1"/> <p>Dates (dd-mm-aaaa):</p> <p>From: <input type="text" value="20-01-2011"/></p> <p>To: <input type="text" value="In progress"/></p> <p>Description:</p> <input type="text"/> </div> <div> <p>Longitude-Latitude: <input type="text" value="40.780541, 4.636230"/></p> <p><input type="button" value="Clear Map"/> <input type="button" value="Save Marker"/></p> <p>Use 'Clear Map' to delete the marker. You have to use the 'Save Marker' button to confirm the marker.</p> <div> </div> </div> <div> Report Information Event <p>Title:</p> <input type="text"/> <p>Type:</p> <input type="text" value="Image"/> <p>Dates (dd-mm-aaaa):</p> <p>From: <input type="text" value="13-02-2011"/></p> <p>Description:</p> <input type="text"/> <p>Select file:</p> <input type="text"/> <input type="button" value="Examinar..."/> <p><input type="button" value="Clear"/> <input type="button" value="Save Resource"/></p> </div> </div>									

Figura 63. Pantalla Guardar Reporte.

Esta pantalla se utiliza para guardar un reporte asociado a una incidencia. Cada reporte debe tener un título, tipo, fecha, descripción y archivo (columna derecha). También es necesario indicar la ubicación del reporte, y sólo podrá ser un punto.

En la columna izquierda aparecen los datos de la incidencia a la que se asocia la acción.

Detalle

General **Detail** Comparison Submit Event Submit Action Submit Report Manage Events Users Help Sign out

**A
C
T
I
O
N
S**

**D
O
S
S
I
E
R**

Event Information
Title:

Type:

Subtype:

Dates (dd-mm-aaaa):
From:
To:
Description:

Filter by date:
Filter: **Highlight:**

Title: 1
Init Date: 10-02-2011
End Date: In progress
Type: Tipo 1
Subtype: SubTipo 1.1

Figura 64. Pantalla Detalle de Acciones.

Esta pantalla se utiliza para visualizar todas las acciones asociadas a una incidencia. Los datos de la incidencia a la que pertenecen todas las acciones aparecen en la columna izquierda. En la columna derecha y en el mapa aparecen todas las acciones.

General **Detail** Comparison Submit Event Submit Action Submit Report Manage Events Users Help Sign out

ACTIONS

DOSSIER

Event Information

Title:

Type:

Subtype:

Dates (dd-mm-aaaa):
From:
To:

Description:

1

Filter by date: Filter: Highlight:

Timeline © SIMILE

2011 Feb 6 Feb 13 Feb 20

Mapa Satélite Relieve

Europe Asia North America South America Africa Australia Indian Ocean Pacific Ocean Atlantic Ocean

POWERED BY Google

TERMINOS DE USO

Report Information Event

Title:

Report Type:

Date (dd-mm-aaaa):

Description:

Figura 65. Pantalla Detalle de Reportes.

Esta pantalla se utiliza para visualizar todos los reportes asociados a una incidencia. Los datos de la incidencia a la que pertenecen todos los reportes aparecen en la columna izquierda. En la parte superior aparece un visor de reportes en la que se pre visualizan las fotos, videos, documentos u otros elementos.

Administrar Incidencias, acciones y reportes.

General	Detail	Comparison	Submit Event	Submit Action	Submit Report	Manage Events	Users	Help	Sign out
Event map - Event list									
Event List									
Id	Title	Type	Subtype	From	Actions				
10	1bis	Tipo 1	SubTipo 1.1	03-02-2011	Delete Event Edit Event				
6	2	Tipo 1	SubTipo 1.1	20-01-2011	Delete Event Edit Event				
11	111111	Tipo 1	SubTipo 1.1	10-02-2011	Delete Event Edit Event				
Actions List									
Id	Main Id	Title	Type	Subtype	From	Actions			
2	6	1	Tipo 1	SubTipo 1.1	10-02-2011	Delete Action Edit Action			
Reports List									
Id	Main Id	Title	Type	From	Actions				
1	6	1	Image	10-02-2011	Delete Report Edit Report				

Figura 66. Pantalla Administrar Incidencias, acciones y reportes.

En esta pantalla aparecen todas las incidencias, acciones y reportes y sirve para poder editar sus datos o eliminarlos.

Administrar Usuarios

General	Detail	Comparison	Submit Event	Submit Action	Submit Report	Manage Events	Users	Help	Sign out
User List								Submit User	
User	Profile	Name	Surname	Phone number	E-mail			Actions	
sigi	Perfil 1	Javier	Prieto Canela	123456789	100047539@alumnos.uc3m.es			Delete User Edit User	

Figura 67. Pantalla Administrar Usuarios.

En esta pantalla aparecen todos los usuarios y sirve para poder editar sus datos o eliminarlos. También se pueden añadir nuevos usuarios.